Nos. 2015-1470, -1554, -1556

IN THE
UNITED STATES COURT OF APPEALS
FOR THE FEDERAL CIRCUIT

MENTOR GRAPHICS CORPORATION, an Oregon corporation,
Plaintiff - Cross-Appellant,

v.

EVE-USA, INC., a Delaware corporation, SYNOPSYS EMULATION AND VERIFICATION S.A.S., formed under the laws of France, SYNOPSYS, INC., a Delaware corporation,
Defendants-Appellants.

APPEALS FROM THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF OREGON
CASES NOS. 3:10-CV-00954-MO (LEAD), 3:12-CV-01500-MO, 3:13-CV-00579-MO, JUDGE MICHAEL W. MOSMAN

BRIEF AND ADDENDUM OF APPELLEE AND CROSS-APPELLANT MENTOR GRAPHICS CORPORATION

George A. Riley
Mark E. Miller
Anne E. Huffsmith
O'Melveny & Myers LLP
Two Embarcadero Center, 28th Floor
San Francisco, California 94111
Telephone: (415) 984-8700

Susan Roeder
O'Melveny & Myers LLP
2765 Sand Hill Road
Menlo Park, CA 94025
Telephone: (650) 473-2600

CERTIFICATE OF INTEREST

Counsel for Plaintiff-Cross-Appellant Mentor Graphics Corporation certifies the following:

1. The full name of every party or amicus represented by me is:

Mentor Graphics Corporation.

2. The name of the real party in interest (if the party named in the caption is not the real party in interest) represented by me is:

Mentor Graphics Corporation.

3. All parent corporations and any publicly held companies that own 10 percent or more of the stock of the party or amicus curiae represented by me are:

None.

4. The names of all law firms and the partners or associates that appeared for the party or amicus now represented by me in the trial court or agency or are expected to appear in this Court are:

O'MELVENY & MYERS LLP: George A. Riley, Mark E. Miller, Luann L. Simmons, Geoffrey H. Yost, Anne E. Huffsmith, Susan Roeder, Elysa Q. Wan, Marc J. Pensabene, Michael A. Koplow (no longer with firm), Nick Janda (no longer with firm), Kristin Hall (no longer with firm), Elizabeth Offen-Brown Klein (no longer with firm), Michael Sapoznikow (no longer with firm), Robin M. Wall (no longer with firm).

KLARQUIST SPARKMAN, LLP: James E. Geringer, Mark W. Wilson, Patrick M. Bible, Salumeh R. Loesch, Andrew M. Mason, Kristin L. Cleveland.

VENABLE, LLP: Megan S. Woodworth.

Dated: September 30, 2015

/s/ Mark E. Miller
Mark E. Miller
Counsel for Plaintiff-Cross-Appellant

TABLE OF CONTENTS

TABLE OF CONTENTS
(continued)

Page

TABLE OF CONTENTS
(continued)

Page

TABLE OF CONTENTS
(continued)

Page

TABLE OF CONTENTS
(continued)

Page

TABLE OF AUTHORITIES

Page(s)

Cases

TABLE OF AUTHORITIES
(continued)

Page(s)

TABLE OF AUTHORITIES
(continued)

Page(s)

TABLE OF AUTHORITIES
(continued)

Page(s)

TABLE OF AUTHORITIES
(continued)

Page(s)

TABLE OF AUTHORITIES
(continued)

Page(s)

TABLE OF AUTHORITIES
(continued)

Page(s)

Statutes

Other Authorities

TABLE OF AUTHORITIES
(continued)

Page(s)

TABLE OF ABBREVIATIONS

'109 Patent	Synopsys' U.S. Patent No. 6,132,109
'176 Patent	Mentor's U.S. Patent No. 5,649,176
'376 Patent	Mentor's U.S. Patent No. 6,240,376
'526 Patent	Synopsys' U.S. Patent No. 7,069,526
'531 Patent	Mentor's U.S. Patent No. 6,009,531
'882 Patent	Mentor's U.S. Patent No. 6,947,882
'962 Patent	Mentor's U.S. Patent No. 6,876, 962
AB	Amicus Brief, filed July 24, 2015
EDA	Electronic Design Automation
EVE	EVE-USA, Inc. and Synopsys Emulation and Verification S.A.S.
FPGA	Field Programmable Gate Array
IPR	Inter Partes Review
Mentor	Mentor Graphics Corporation
PTAB	Patent Trial and Appeal Board
SPB	Synopsys' Principal Brief, filed July 17, 2015
Synopsys	Synopsys, Inc., EVE-USA, Inc. and Synopsys Emulation and Verification S.A.S.
ZeBu emulators	ZeBu Server, ZeBu Server 2, ZeBu Server 3, and ZeBu Blade emulators

STATEMENT OF RELATED CASES

Mentor agrees with the statement of related cases provided by Synopsys, with the following clarifications and additions:

1. In the IPR of the '376 Patent (SPB xiii-xiv), the PTAB confirmed the patentability of all claims at issue in this appeal. Oral argument in the appeals from the Board decision has been set for November 3, 2015.

2. Oral argument in the appeal of Synopsys' APA action (SPB xv) has also been set for November 3, 2015.

3. All asserted claims of a fourth patent asserted by Synopsys in the Northern District of California were found to be unpatentable in a June 11, 2015 final written decision of the PTAB. Synopsys has appealed the Board decision to this Court. *Synopsys, Inc. v Mentor Graphics Corp.*, No. 15-2056 (Fed. Cir.).

JURISDICTIONAL STATEMENT

Mentor timely cross-appealed the November 14, 2014 Final Judgment following the district court's ruling on Synopsys' post-trial motions. A188; A203; A36443-43. This Court has jurisdiction over both parties' appeals from the final judgment, although issues concerning Mentor's supplemental damages remain before the district court. 28 U.S.C. § 1292(c)(2); 28 U.S.C. § 1295(a)(1). Synopsys invokes this Court's jurisdiction under 28 U.S.C. § 1292(a)(1) and (c)(1) over its separate appeal of the permanent injunction, but Synopsys has waived any

arguments concerning the injunction by failing to raise them in its principal brief. *See SmithKline Beecham Corp. v. Apotex Corp.*, 439 F.3d 1312, 1319-20 (Fed. Cir. 2006).

STATEMENT OF THE ISSUES

Synopsys' Appeal

1. Whether substantial evidence supports the jury's verdict that Synopsys' ZeBu emulators infringe Claims 1, 24, 26, 27, and 28 of the '376 Patent using the ordinary meaning of "indicative" and "identifying."

2. Whether a patentee who demonstrates that "but for" the infringement it would have made additional sales of its product is entitled to recover all of the profits it lost, or only a portion of the profits it lost absent satisfaction of the entire market value rule ("EMVR").

3. Whether the district court properly applied assignor estoppel, where the inventor of the '376 Patent assigned the patent to Mentor then proceeded to actively assist Synopsys in infringing the patent.

4. Whether the district court correctly held that the '109 Patent is indefinite for failing to disclose, with reasonable certainty, the scope of the claim term "near," where the claims, the specification, and the prosecution history fail to address the meaning or scope of "near."

5. Whether the district court correctly held that the claims of the ’526 Patent are invalid under 35 U.S.C. § 101 and *In re Nuijten*, 500 F.3d 1346 (Fed. Cir. 2007), because the claims encompass transitory, propagating signals.

Mentor’s Cross-Appeal

1. Whether the district court erred in excluding Mentor’s willfulness claim, where Synopsys acquired an entity (EVE) selling the accused product under a license from Mentor and continued selling the product despite termination of the license, simply because Synopsys and EVE preemptively filed a declaratory judgment action before the license terminated.

2. Whether the district court erred in holding Claims 7, 9, and 13 of the ’882 Patent invalid for failure to satisfy the written description requirement, where the claim language alleged to lack support was present in the specification as originally filed, and the district court imposed a requirement on the written description that is inconsistent with the district court’s claim construction.

3. Whether the district court erred in determining that res judicata barred Mentor’s claims that Synopsys infringed the ’176 and ’531 Patents, where Mentor’s claims were based exclusively on acts of infringement that occurred long after resolution of the prior litigation between the parties.

STATEMENT OF THE CASE

I. PATENTS AT ISSUE

When a circuit design is loaded into logic devices (FPGAs) in an emulator, many of the signals are located deep within the FPGAs and are not available at the output pins of the FPGAs. The inventions claimed in the ’376 Patent involve the creation of “instrumentation signals” (claims 1 and 28) or the generation of logic (claim 24) to provide visibility into circuit designs being emulated in order to locate and identify errors. A532-33. Contrary to Synopsys’ statements (SPB 4), the asserted claims do not require “instrumentation data.” *Id*. Synopsys also conflates two embodiments of the invention—one adds instrumentation signals or logic to a design and the other uses a cross-reference database. A525(2:40-55). The use of a cross-reference database is not at issue.

In addition to the patents discussed by Synopsys—Mentor’s ’376 Patent and Synopsys’ ’109 and ’526 Patents—Mentor’s cross-appeal relates to three Mentor patents: the ’882, ’531, and ’176 Patents. The ’882 Patent is described below. The substance of the ’531 and ’176 Patents is not material to this appeal because the district court dismissed Mentor’s claims of infringement of those patents on procedural grounds (i.e., res judicata).

The ’882 Patent relates to the clocks that control the operation of logic devices and the interconnections that transfer signals between the logic devices in

an emulator. A552(Abstract). The logic devices have a limited number of pins which can create bottlenecks when transferring signals between the logic devices through the limited number of pins. A565. In one embodiment of the inventions disclosed and claimed in the ’882 Patent, a first clock and a second clock control the operation of the logic devices, and a signal routing clock, that operates at a higher frequency than the first and second clocks but is otherwise independent of the first and second clocks, controls the operation of the interconnections. A566(4:16-28). The use of the faster signal routing clock to control the transfers of data between logic devices speeds transfers and eliminates bottlenecks. A566(4:24-36).

II. TIMELINE

A. Burgun Invented The Debugging Technology Claimed In The ’376 Patent While Employed By Mentor, Then Left To Found EVE (1998-2000)

Dr. Luc Burgun and Dr. Alain Raynaud, the two named inventors of the ’376 Patent, invented the technology claimed in the ’376 Patent in 1998 while employed by Mentor. A1571-614. The ’376 Patent claims methods for debugging a circuit design using, for example, an emulator. A1581-614. At trial, Dr. Burgun testified that debugging a circuit design is “the main purpose of an emulator.” A41672:6-9; A41551:11. Dr. Burgun admitted that “through the ’376 Patent,[he

was] trying to improve the debug capabilities of emulators." A41660:22-24. In 2000, Dr. Burgun left Mentor to found EVE. A1616-29.

B. Mentor Sued EVE For Infringing Its '376, '176, And '531 Patents, And EVE Settled By Taking A License (2006)

In 2006, Mentor sued EVE alleging infringement of the '376, '176 and '531 Patents. A1658-64 (the "2006 Action"). The parties settled the litigation in December 2006 and Mentor licensed EVE under the '376, '176, and '531 Patents. A3659-62 ("Mentor-EVE License"). The Mentor-EVE License included a prohibition on assignment: "This license does not include the right to sublicense, and terminates if a controlling interest in EVE is acquired by any other company in the EDA industry." A3661.

C. EVE Developed The Infringing Technology Under The License In Order To Compete In The Emulation Market (2009)

EVE's former Vice President of Marketing described EVE's early emulators as "not competitive." A41716:2-24; A41718:23-25. He confirmed that "it really wasn't until 2009 when the ZeBu Server [emulator] was announced that EVE began to be competitive." A41716:25-A41717:2.

The critical weakness of EVE's early emulators was their poor debugging functionality, also referred to as "debug visibility." Several of EVE's customers and potential customers, including EVE's largest customer, Intel, complained about the inability of EVE's emulators to effectively debug circuit designs and

demanded improvements. A41718:1-4; A41672:13-A41673:2; A41717:9-15; A41721:20-22.

Internal EVE emails from 2007-2009 show that the company faced a loss of key customers if it could not improve debug visibility. An email reported "BIG trouble" because Intel had placed a hold on Eve "orders pending the results of a competitive eval," EVE's lack of "'debug' was the big issue," and "[i]t would be great to deliver a credible debug solution …." A44415-17. Alain Raynaud responded that "[t]he key feature … needed for ZeBu to beat [Mentor's] Veloce is full visibility." A44414. At MIPS, EVE's "50% pricing" and ability to "meet[] speed requirements" were insufficient to displace Mentor as MIPS' emulator supplier, based in large part on EVE's lack of "full visibility." A44419-20. And Hi-Silicon chose Mentor over EVE based on Mentor's "better debug facility," listing issues related to EVE's debug facility as "the deal breaker." A44467-69.

In response to these complaints, EVE, operating under the Mentor-EVE License, developed the infringing flexible probes and value change probes. A41672:13-A41673:2; A41717:9-15; A41721:20-22. By adding Mentor's patented debugging technology to the ZeBu emulators, EVE began to compete with Mentor and Cadence for emulator sales. Several of Synopsys' trial witnesses confirmed that the infringing flexible probes and value change probes were essential to EVE's success in the marketplace. *See* A41714:11-14 ("before the

development of flexible probes in 2009, EVE was not competitive."); A41761:5-9 ("Without [debug capabilities], [EVE] would have horrible times. With that, [EVE] became competitive."); *see also* A41723:11-24; A41757:12-14; A41761:5-9; A41673:12-16; A42188:7-12.

D. Mentor Brought Suit Alleging EVE Infringed Its '962 And '882 Patents (2010-2012)

In August 2010, Mentor sued EVE alleging infringement of the '962 Patent and, in August 2012, Mentor sued EVE alleging infringement of the '882 Patent. A1300-11.

E. Mentor Notified Synopsys That EVE's License To Practice The '376, '176, And '531 Patents Would Terminate If Synopsys Acquired EVE (2012)

On August 20, 2012, in response to indications that Synopsys was planning to acquire EVE, Mentor's CEO sent an email to Synopsys' CEO offering to waive the confidentiality provision of the Mentor-EVE License to ensure that Synopsys was aware that the Mentor-EVE License would not survive the acquisition. A40779:15-17; A43919. Shortly thereafter, Synopsys' President and COO expressed reluctance to complete the acquisition because "[t]he legal issues we will inherit is an extra risk of a different dimension that we are taking on." A44133. And Synopsys informed EVE during acquisition negotiations that, in light of Synopsys' assumption of the risk of "both known and unknown IP claims," the

purchase price for EVE would have to be reduced. A41627:23-A41628:9; A43918.

F. Synopsys Sued Mentor And Acquired EVE, Terminating EVE's License (2012)

On September 27, 2012, anticipating the termination of EVE's license, Synopsys filed a declaratory judgment action against Mentor relating to the '376, '176, and '531 Patents. A1200-09. On October 4, 2012, Synopsys completed its acquisition of EVE, thereby terminating EVE's license under the '376 Patent and EVE's ability to legally make, use, and sell ZeBu emulators including flexible probes and value change probes (the accused features). A1849-50. Despite termination of the license, Synopsys continued to sell ZeBu emulators. Mentor filed counterclaims on January 11, 2013, alleging willful infringement of the '376, '176, and '531 Patents. A1216-29. On June 4, 2013, Synopsys amended its complaint, adding allegations that Mentor's emulators infringed the '526 and '109 Patents. A3291-303.

G. The Cases Were Consolidated And Litigated In District Court (2012-2014)

The district court consolidated Synopsys' declaratory judgment action and suit for infringement of the '526 and '109 Patents with Mentor's suits for infringement of the '962 and '882 Patents. A1500. On summary judgment, the court removed all but Mentor's '376 Patent from the case. A121-22; A14060-62;

A23748-56. Shortly before trial, the court granted Synopsys' motion *in limine* to bar Mentor's willfulness claim. A26720; A137. At trial, a jury found that Synopsys' ZeBu emulators infringed Claims 1, 24, 26, 27, and 28 of the '376 Patent and awarded $36,659,771.45 in lost profits and royalties. A185-87.

Despite the jury verdict and entry of final judgment, Synopsys continued to sell infringing ZeBu emulators. The district court affirmed the jury's findings of infringement and damages award and entered a permanent injunction against future infringement. A193-203; A204-07. In granting the permanent injunction, the district court explained:

> There is no justification for these sales. Synopsys knew its ZeBu emulator had been found to infringe the '376 patent, and should have ceased selling that emulator. Harm arising out of Synopsys's willful illegal behavior post-verdict cannot justify denying an injunction ….

A36447-48.

SUMMARY OF ARGUMENT

Synopsys' Appeal

I. Synopsys agreed that the ordinary meaning of "indicative" and "identifying" were to be used by the jury. There is no basis in the language of the '376 Patent for a definition of those terms that would require ZeBu's instrumentation signals to "directly" or "automatically" "specify those very lines" of source code that produced the signals. Substantial evidence supported the jury's verdict that the instrumentation signals produced by the ZeBu emulators meet the

"indicative" and "identifying" limitations because those signals include the RTL name of the corresponding source code and show the execution status of that source code (Claims 1 and 28) and whether it was active during simulation (Claims 24, 26, and 27).

II. The Supreme Court has made clear that, to determine the amount of damages to which an injured patentee is entitled, "we must ask how much [the patentee] suffered by [the infringer's] infringement—how much it would have made if [the infringer] had not infringed." *Aro Mfg. Co. v. Convertible Top Replacement Co.*, 377 U.S. 475, 507 (1964). Where an injured patentee demonstrates that it would have made additional sales "but for" the infringement, as Mentor did in this case and which Synopsys does not dispute on appeal, the injured patentee is entitled to lost profits damages in an amount equal to the entire amount of the profits the patentee lost on those additional, lost sales, not some lesser portion of the profits the patentee lost.

III. Dr. Burgun assigned the '376 Patent to Mentor, left to form EVE, took a license from Mentor to enable EVE to make otherwise infringing emulators, then sold EVE to Synopsys and assisted Synopsys in continuing to manufacture and sell those emulators despite termination of the Mentor-EVE License. Under this Court's precedent, such conduct triggers application of the assignor estoppel doctrine. *See Diamond Scientific Co. v. Ambico, Inc.*, 848 F.2d 1220, 1223-24

(Fed. Cir. 1988). The Supreme Court's decision in *Lear, Inc. v. Adkins*, 395 U.S. 653 (1969), did not abolish assignor estoppel, and this case presents no reason to revisit the doctrine.

IV. The district court correctly held that the '109 Patent does not inform, with reasonable certainty, a person of ordinary skill of the scope of "near." The '109 Patent requires that items are displayed "near" each other, but the claims, specification, and prosecution history never discuss or define the term "near." Synopsys' proposed interpretation of "near" is subjective and demonstrates the uncertainty of the term.

V. All claims of Synopsys' '526 Patent are directed to a "machine-readable medium," which the specification explicitly defines to include carrier waves. Because carrier waves are unpatentable, transitory signals, *In re Nuijten*, 500 F.3d at 1354, the claims encompass patent-ineligible subject matter under 35 U.S.C. § 101 and are invalid.

Mentor's Cross-Appeal

VI. The district court erred in rejecting Mentor's willfulness claim on the objective prong by granting Synopsys' pretrial motion *in limine*. The district court's decision was based solely on the procedural posture of the case rather than consideration of the totality of circumstances or an assessment of the reasonableness of Synopsys' failed infringement defenses. The district court

erroneously (1) refused to consider evidence that Synopsys acquired EVE for the very purpose of infringing the ’376 Patent; (2) used the date of Synopsys’ filing of a declaratory judgment action—rather than the date more than three months later when Mentor filed its willfulness counterclaim—as the cutoff date between “prelitigation” and “postlitigation” willfulness; and (3) disqualified Mentor from basing its willfulness claim on any “postlitigation” conduct because Mentor did not seek a preliminary injunction, even though this Court has not enunciated such a categorical rule.

VII. The district court erred in holding that Mentor’s ’882 Patent fails to satisfy the written description requirement. Claim 5 requires “independent” clocking, which the district court construed to mean “no required timing relationship between clock edges.” Yet, in holding the ’882 Patent invalid, the district court applied a more restrictive interpretation of “independent,” prohibiting any relationship, either a frequency relationship or a clock edge relationship. The district court then held that the specification fails to demonstrate possession of “independent” clocking because the specification discloses a relationship between clock frequencies. A relationship between clock frequencies is not a “required timing relationship between clock edges” and the specification explicitly excepts a difference in frequencies from the determination of independence. Finally, the inventors had possession of the concepts covered by claim 5 at the time of filing

the application for the ’882 Patent—original claim 1 included the same language that is at issue in claim 5.

VIII. Mentor’s claims that Synopsys infringed the ’176 and ’531 Patents were based exclusively on acts of infringement that occurred after the 2006 Action was dismissed. Indeed, Mentor’s claims alleged infringement of the ’176 and ’531 Patents commencing on October 4, 2012, after the Mentor-EVE License terminated. A prior judgment on the merits “cannot be given the effect of extinguishing claims which did not even then exist and which could not possibly have been sued upon in the previous case.” *Lawlor v. Nat’l Screen Serv. Corp.*, 349 U.S. 322, 328 (1955). Because Mentor’s infringement claims did not exist at the time of the 2006 Action, they cannot be barred by res judicata.

ARGUMENT IN OPPOSITION TO SYNOPSYS’ APPEAL

I. THE DISTRICT COURT CORRECTLY UPHELD THE JURY’S FINDING OF INFRINGEMENT

Synopsys focuses on one limitation in each claim that the jury found was infringed: the requirement in Claims 1 and 28 that the instrumentation signals in a netlist “are indicative of an execution status” and the requirement in Claim 24 of “identifying a process as active during simulation.” SPB 22. Synopsys does not dispute that ZeBu emulators practice all other elements of those claims.

Claim 1 requires two steps: (a) “identifying at least one statement within a register transfer level (RTL) synthesizable source code,” and (b) “synthesizing the

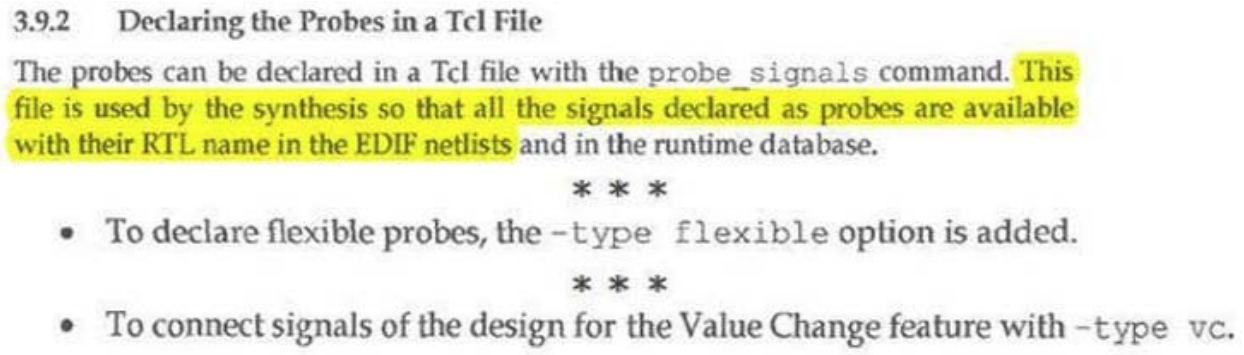
source code into a gate-level netlist including at least one instrumentation signal" that is "indicative of an execution status of the at least one statement" of RTL source code. A532(15:2-8). Claim 28 requires the same synthesis step as Claim 1. 533(17:61-18:7).

Claim 24 requires three steps: (a) "identifying a sensitivity list of a process"; (b) "generating logic to identify signal events" for any signal in the sensitivity list; and (c) "identifying the process as active during simulation when a signal event occurs for any signal in the sensitivity list."[1] A533(17:37-44).

A. Substantial Evidence Supported The Jury's Determination That Zebu Emulators Infringe

Substantial evidence supported the jury's determination that ZeBu emulators "indicate" and "identify" RTL source code statements within the ordinary meaning of those terms. Mentor's expert, Dr. Sarrafzadeh, explained to the jury that a chip designer using a ZeBu emulator selects one or more RTL source code statements to debug. A41127:15-21. The ZeBu manual walks the chip designer through the process of inserting flexible probes and value change probes to debug those statements. A43212(§ 3.9.2.) The designer creates a file, called a "tcl" file, in which the probes are specified using the "probe_signals" command:

[1] The jury also found infringement of Claims 26 and 27, which depend from Claim 24. A182-83; A533. For purposes of this brief, Mentor's discussion of Claim 24 also applies to Claims 26 and 27.



3.9.2 Declaring the Probes in a Tcl File

The probes can be declared in a Tcl file with the `probe_signals` command. This file is used by the synthesis so that all the signals declared as probes are available with their RTL name in the EDIF netlists and in the runtime database.

* * *

- To declare flexible probes, the `-type flexible` option is added.

* * *

- To connect signals of the design for the Value Change feature with `-type vc`.

A43212-13(highlighted).[2]

The "probe_signals" command identifies a portion of the RTL source code by name, creating a "clear association" between the name of the probe and the source code that is being probed. *Id.*; A42417:1-22. In the example presented to the jury, to debug an RTL source code process named "mux2to1," which relates to a multiplexer, the user would create a tcl file with the command "probe_signals" followed by "mux2to1" as well as terminology specifying a flexible or value change probe. A41130:7-14; *see also* A41828:22-23 (admission by Synopsys' research and development group director that ZeBu "keep[s] the name in the design").

During synthesis, the probe_signals command causes the creation of additional logic known as a tracer circuit. A41128:7-A41129:14. The tracer circuits output signals based on the operation of the original logic (the circuit being

[2] Mentor's expert testified that flexible probes and value change probes use the same probe signals process and have the same result, and Synopsys does not argue otherwise. *See*, *e.g.*, A41120:11-25; A42445:1-8.

debugged) and "indicative" of whether the corresponding RTL source code statement(s) is active. A41139:10-21. Those signals are represented by computer code in a netlist, which is a list of logic elements and their interconnections. A41089:14-24; A41128:7-14; A156. The ZeBu manual confirms that ZeBu performs synthesis to create a netlist and that "all the signals declared as probes are available with their RTL names in the ... netlist…." A43212(§ 3.9.2). Thus, ZeBu's netlist correlates the results of emulation back to the RTL source code being debugged, which informs the designer as to whether the source code statement(s) are active, as required in Claims 1 and 28. A42437:15-A42438:24. In the example where the source code for the "mux2to1" source code is being debugged, the instrumentation signals generated during synthesis are indicative of the execution status of the "mux2to1" source code, i.e., whether that source code is active. *Id.*; A41112:19-A41113:12.

Dr. Sarrafzadeh also described how the ZeBu emulators infringe Claim 24 using the tracer circuits created by flexible and value change probes. Synopsys does not dispute that the ZeBu emulators identify a sensitivity list of a process. (A "process" is "a description of the behavior of some portion of a circuit design," i.e., a portion of the source code, A156; A42425:24-42426:2, and a sensitivity list is "a list of signals to which a process is responsive," i.e., a list of the inputs and outputs to that portion of the source code, A156; A41133:1-20.) Synopsys does

not dispute that the ZeBu emulators generate logic (tracer circuits) that identify signal events for signals in the sensitivity list. A42427:13-A42428:1; A41134:12-25. A change in the signal output by the tracer circuit (which is available with the RTL name) associated with a signal in the sensitivity list means there has been an event on that process (portion of the source code). A41133:6-A41136:17; A42418:16-A42423:18; A42427:13-A42428:13; A42431:16-A42432:16. A signal event for any of the signals in the sensitivity list identifies the process as active. *Id*.

ZeBu emulators thus "connect 'signal values'" to "particular source code lines," which Synopsys contends is the "premise" of the '376 Patent. SPB 34. Because there was substantial evidence that ZeBu emulators "indicat[e]" and "identify[]" the execution status of particular RTL statements, the district court correctly upheld the jury's infringement verdict. A199; A42611:23-42612:11; *see Revolution Eyewear, Inc. v. Aspex Eyewear, Inc.*, 563 F.3d 1358, 1370-71 (Fed. Cir. 2009) (jury's verdict must be upheld if supported by substantial evidence, i.e., relevant evidence that reasonable minds might accept as adequate to support the jury's conclusion).

B. "indicative" And "identifying" Do Not Require "direct" Or "automatic" Specification Of Particular Lines Of Source Code Or A Particular Process

Nothing in the '376 Patent or the common understanding of "indicative" or "identifying" requires that the signals "directly" or "automatically" specify the particular lines of source code or a particular process. Having failed to ask the district court to construe "indicative" and "identifying," Synopsys cannot now argue that the '376 Patent additionally requires "direct" and "automatic" specification of "those very lines of code" or process that executed. SPB 21, 32, 36.

1. Synopsys' Failure To Ask The District Court To Construe "indicative" Or "identifying" Bars Synopsys From Arguing That "direct" Or "automatic" Action Is Required

Synopsys asks this Court to read requirements for "direct" and "automatic" specification of corresponding source code into the claim terms "indicative" and "identifying." But as Synopsys admits, it agreed that "indicative" and "identifying" were to be given their ordinary meanings, did not ask the district court to construe those terms, and does not ask this Court to do so. SPB 35. Synopsys' agreement makes the district court's post-trial statement regarding claim construction irrelevant. "[W]here the parties and the district court elect to provide the jury only with the claim language itself," it is too late after trial "to argue for or adopt a new and more detailed interpretation of the claim language and test the

jury verdict by that new and more detailed interpretation." *Warsaw Orthopedic, Inc. v. NuVasive, Inc.*, 778 F.3d 1365, 1373 (Fed. Cir. 2015) (quoting *Hewlett-Packard Co. v. Mustek Sys.*, 340 F.3d 1314, 1321 (Fed. Cir. 2003)). Instead, review of the jury's verdict is limited to the question of whether substantial evidence supported the verdict under the applicable jury instruction, which here required the jury to use the ordinary meaning of "indicative" and "identifying." A156.

2. Claims 1, 24, And 28 Do Not Require Signals That "directly" Or "automatically" Specify A Particular Line Of Source Code Or Process

Synopsys attempts to read requirements for "automatic" or "direct" specification of a particular line of source code or process into Claims 1, 24, and 28. But to do so, Synopsys incorrectly focuses on what steps a designer must take to evaluate signals output by a ZeBu emulator—specifically, whether the designer can "determine whether a particular source code statement was executed on the basis of the waveform viewer alone" or whether the designer must "perform extra work to correlate the test result with the specific line(s) of code." SPB 21, 36. Claims 1 and 28, however, do not "***start*** by synthesizing particular source code into a netlist containing an 'instrumentation signal.'" SPB 30 (emphasis added). The creation of a netlist is the ***end result*** of the process of Claims 1 and 28. See A532(15:2-8); A533(17:61-18:7). The signals in the netlist are simply information

in a data file. *See* A41089:14-24. It would be illogical to construe the claims to require that information to "automatically" or "directly" *do* anything. Rather, the netlist need only include a signal that is indicative of the execution status of a statement in the source code. Claims 1 and 28 do not require further steps of loading the netlist into an emulator, displaying results using a waveform viewer, or identifying source code using the netlist. Here, the ZeBu manual establishes that "all the signals declared as probes are available with the RTL name in the [] netlists," thus satisfying Claims 1 and 28. A43212(§ 3.9.2).

Claim 24 requires generating logic to identify signal events in circuits corresponding to a process (a portion of source code) selected for debugging and identifying that process as active during simulation when a signal event occurs. A533(17:37-44). Again, the signals output from the tracer circuits created by the flexible and value change probes show whether the probed process is active. Claim 24 requires nothing more. There is no basis to require a signal to do anything "automatically" or "directly."

Synopsys' argument that the claim terms "indicative" and "identifying" must be read in a way that precludes "extra work to correlate the test result with the specific line(s) of code" (SPB 21) distorts the problem the '376 Patent was trying to solve. The '376 Patent addresses the difficulty of applying "millions or billions of test vectors" in order to adequately test complex designs if "none of the high-

level information available in the RTL source code" is included. A526(4:58-66). This problem is solved by creating instrumentation logic that "provides an output signal indicative of whether the corresponding synthesizable statement is active." A527(5:17-30); *see also* A41102:2-11; A41103:25-A41104:18. ZeBu emulators solve the problem in the manner described in the patent: the designer inserts flexible or value change probes (tracer circuits that output signals containing the RTL name of the corresponding source code statement), and those output signals indicate the execution status of that source code and identify whether it is active. As Dr. Sarrafzadeh explained to the jury, using the RTL name to connect the signal back to the relevant source code is a task that can be performed quickly by using the search function of a text editor. A42426:16-A42427:1; A42423:10-A42424:25; A42425:14-20.

3. The Ordinary Meanings Of "indicative" And "identifying" Do Not Require That An Instrumentation Signal "automatically" Or "directly" Specify A Particular Line Of Source Code Or Process

Requirements for "direct" and "automatic" specification of source code or a particular process are not only absent from the '376 Patent, but they are not required under the "commonly understood meaning of those words." *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1333 (Fed. Cir. 2005) (en banc). "Indicate" means "to point out or point to or toward with more or less exactness" or "show or make known with a fair degree of certainty." *Webster's Third New Int'l Dictionary*

(1986); *see also Oxford English Dictionary* (1989) (defining "indicative" as "that hints or suggests"); *Merriam-Webster's Collegiate Dictionary* (1998) (defining "indicate" as "to be a sign, symptom, or index of"). "Identify" means "to serve as a means of identification for." *Oxford English Dictionary* (1989). Such definitions do not "contradict any definition found in or ascertained by a reading of the patent documents." *Phillips*, 415 F.3d at 1334 (quoting *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1584 n.6 (Fed. Cir. 1996)).

The terms "indicative" and "identifying" in Claims 1, 24, and 28 thus require only that the signals "point out" and "make known" the execution status of the RTL source code statements being debugged and "serve as a means of identification" of whether a process is active. There is no requirement that the signals "automatically" or "directly" specify a particular line of source code or process. The output of ZeBu's emulators is not like Synopsys' examples of "raw data on angles of orientation or gas pressure readings" (for brake or fuel indicators), "the volume and number of molecules of gas in a container" (for an oven temperature display), or "ice-cream-selection principles" (for ordering ice cream). SPB 32-33. ZeBu's tracer circuits (both as represented in a netlist and as implemented in an emulator) output signals that *specify the RTL source code by name*. ZeBu's netlist is therefore analogous to asking the Baskin-Robbins clerk for

vanilla ice cream, which "indicates" and "identifies" what flavor one wants even though the clerk must locate the correct tub of ice cream.

None of the cases cited by Synopsys dictate otherwise. In *Exergen Corp. v. Wal-Mart Stores, Inc.*, this Court rejected a patentee's attempt to change the meaning of "indication" that was "given to the jury." 575 F.3d 1312, 1321 (Fed. Cir. 2009). The patented infrared thermometer required "a display for providing an indication of the internal temperature." *Id.* at 1320. The district court had construed "internal temperature" as "temperature of the region existing beneath the surface of the biological tissue targeted for measurement." *Id.* The patentee's expert and the co-inventor of the patent-in-suit explained to the jury that "providing an indication" meant "the number on the display must itself be the value of the internal temperature." *Id.* at 1321. The accused product targeted a patient's forehead for measurement, and the question on appeal was whether the accused product displayed a reading of the value of the temperature of the temporal artery beneath the skin of the forehead. *Id.* at 1320-21. It was undisputed that the accused product did not do so, but instead displayed the patient's oral temperature (from which the temperature of the forehead temporal artery could be calculated). Based on the unchallenged claim construction and what the patentee itself argued the claim required, the Court rejected the patentee's argument on appeal that the thermometer infringed. *Id.*

Unlike *Exergen*, none of the claims at issue here require the "display" of a particular piece of information, and Mentor's expert, Dr. Sarrafzadeh, never testified that the netlist must automatically or directly display a particular line of RTL source code (as Synopsys is trying to rewrite the claim). Instead, Dr. Sarrafzadeh explained that ZeBu's signals satisfied the claims because the signal name includes the RTL source code name and shows its execution status and whether it was active. *See, e.g.*, A42431:16-A42432:16; A42437:22-A42438:9. If anything, *Exergen* supports Mentor's position that a party may not on appeal change the meaning of a term that was given to the jury.

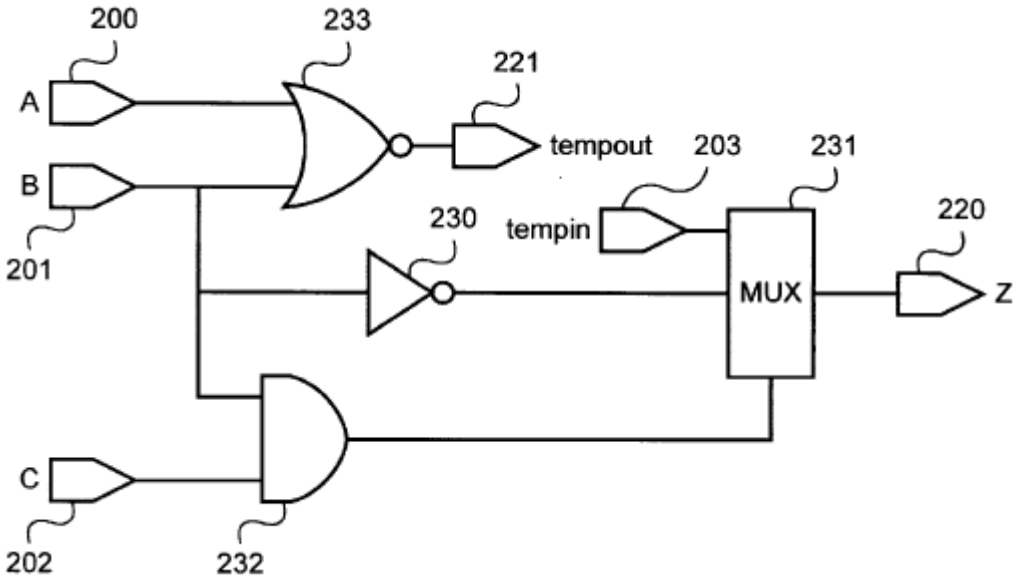
In *In re Bookstaff*, 606 F. App'x 996 (Fed. Cir. 2015) (non-precedential), also cited by Synopsys, a total value transmitted by a credit card issuer was not "indicative of the gratuity to be charged" because "there [was] no way from the data transmitted by the card issuer to ascertain the amount of the total that is indicative of the gratuity" and that total value "must be compared to the original bill amount presumably known by the patron to determine the gratuity." Here, by contrast, the instrumentation signal includes the name of the RTL code that generated the signal and provides the indication required by the claims. *See, e.g.*, A42423:13-18; A43212(§ 3.9.2).

Finally, Dr. Sarrafzadeh testified there was a "clear association" between the name of the probe and the RTL source code. A42417:15-22. This satisfies the

dictionary definition of "identify" cited in *Gemstar-TV Guide, Int'l, Inc. v. ITC*, i.e., "to link in an inseparable fashion: make correlative with something." 383 F.3d 1352, 1367 (Fed. Cir. 2004).

4. Mentor Did Not Disclaim Constructions Of "indicative" Or "identifying" That Require Source Code Identification Effort

Synopsys has not met its high burden of showing that Mentor "unequivocally and unambiguously disavowed" the scope of "indicative of" or "identifying" by making statements about the prior art Gregory patent during IPR proceedings. *See Biogen Idec, Inc. v. GlaxoSmithKline LLC*, 713 F.3d 1090, 1095 (Fed. Cir. 2013). Synopsys argued in the IPR proceedings that the "tempout" signal in the following Figure 9 from Gregory is an "instrumentation signal" that is "indicative of an execution status" and thus anticipated Claims 1 and 28 of the '376 Patent:



A465; A26350; A26360. Figure 9 corresponds to the source code in Figure 4 of Gregory, which Synopsys reproduces in its brief:

```
if (C and B) then
        Z <= not(A or B);
else
        Z <= not B;
end if;
```

SPB 8; A460; A494.

Mentor disagreed with the assertion that the "tempout" signal was the same thing as an "instrumentation signal" that is "indicative of execution status." Mentor explained that "tempout is not indicative of the execution status of the probed statement" because while tempout provides the result of "not(A or B)," this does not depend on the condition "(C and B)." The condition of "(C and B)," not the result of "not(A or B)," determines whether or not the probed statement is

executed. A26419-20. Gregory's signals would not enable a user to determine whether the condition "(C and B)" was executed. Thus, Gregory's signals are not, by their nature, "indicative of execution status," and Gregory did not anticipate Claims 1 and 28. A26417-21. Mentor did not address signal analysis in Gregory, much less make any argument about "manual" versus "automatic" determinations of execution status. *Id.*

Also, Mentor's IPR statement that Gregory did not disclose the "identifying … a signal event" step of Claim 24 did not address whether that claim reads on ZeBu's signals. In the IPR petition, Synopsys argued that a particular signal in Gregory "can be used to identify when a signal event occurs during simulation." A26359. Mentor simply pointed out that Gregory failed to disclose using those signals for that purpose. A26375-76.

The ZeBu emulators do not practice the Gregory patent (SPB 16) and, unlike the signals in Gregory, ZeBu emulators provide information indicative of execution status and identify whether there has been activity on the circuit that is being probed. Thus, as the district court held repeatedly, Mentor's statements about Gregory are not a disclaimer of coverage over Synopsys' infringing products. *See, e.g.*, A42611:23-A42612:11.

Accordingly, this Court should affirm the judgment of infringement.

II. THE DISTRICT COURT CORRECTLY UPHELD THE JURY'S DAMAGES AWARD

At trial, Mentor proved that, "but for" Synopsys' infringement, Mentor would have made additional sales of its Veloce emulator. A164-71, A182-84. Mentor demonstrated "but for" causation by satisfying the universally accepted *Panduit* test. *Id.* "A showing under *Panduit* permits a court to reasonably infer that the lost profits claimed were in fact caused by the infringing sales, thus establishing a patentee's prima facie case with respect to 'but for' causation." *Rite-Hite Corp. v. Kelley Co.*, 56 F.3d 1538, 1545 (Fed. Cir. 1995). When a patentee establishes that there was a reasonable probability that the sales would have been made "but for" the infringement, "e.g., by satisfying the *Panduit* test, it has sustained the burden of proving entitlement to lost profits due to the infringing sales." *Id.*

Synopsys does not challenge Mentor's proof of "but for" causation. SPB 43-58. Indeed, Synopsys does not raise any issues related to any of the *Panduit* factors. *Id.* Synopsys argues, instead, that, despite Mentor's proof of "but for" causation, Mentor is entitled to only a portion of the profits it lost as a result of Synopsys' infringement.[3] *Id.* Synopsys' position is not supported by the law and would produce a manifestly unjust result.

[3] The district court made clear, pre-trial, that "if Mentor Graphics is successful proving "but for" causation, … Mentor Graphics is not obligated to show

A. Apportioned Lost Profits Can Never Adequately Compensate A Patentee For Sales Lost As A Result Of Infringement

In 1946, Congress amended 35 U.S.C. § 284, which governs recovery in infringement actions. *See Gen. Motors Corp. v. Devex Corp.*, 461 U.S. 648, 652 (1983). Amended section 284 provides that "[u]pon finding for the claimant the court shall award the claimant damages ***adequate to compensate for the infringement*** ...." 35 U.S.C. § 284 (emphasis added). Through this amendment, "Congress sought to ensure that the patent[ee] would in fact receive ***full compensation*** for '***any damages***' he suffered as a result of the infringement." *Gen. Motors*, 461 U.S. at 654-55 (emphasis added) (citing H.R. Rep. No. 1587, 79th Cong., 2d Sess. 1-2, 1 (1946) ("any damages the complainant can prove"); S. Rep. No. 1503, 79th Cong., 2d Sess. 2 (1946), U.S. Code Cong. Serv. 1946, at 1387 (same)). To determine the amount of this "full compensation," "we must ask how much [the patentee] suffered by [the infringer's] infringement—***how much it would have made if [the infringer] had not infringed***." *Aro Mfg.*, 377 U.S. at 507 (emphasis added); *see also id.* (quoting *Yale Lock Mfg. Co. v. Sargent*, 117 U.S. 536, 552 (1886)) (damages should "'constitute the difference between [the patent

---

apportionment." A40538:20-22. This ruling stood throughout the trial, despite Synopsys' repeated requests for reconsideration during formulation of the jury instructions. *See, e.g.*, A40664:24-A40665:21, A41405:18-A41416:7. Synopsys' characterization of the district court as having "careened from one position to another" is unfounded. SPB 18. Regardless, the district court's careful consideration of this issue has no impact on this Court's de novo review.

owner's] pecuniary condition after the infringement, and what his condition would have been if the infringement had not occurred'").

Where a patentee proves that, "but for" the infringement, it would have made additional sales of its product—as Mentor proved at trial and which Synopsys does not dispute on appeal—the patentee can only be fully compensated through a damages award that reflects the entire amount of the profits it lost on those additional, lost sales. Had the infringement not occurred, the patentee would have made its full profits on every lost sale, not some lesser portion of its profits.

This is not, as Synopsys argues, an unreasonable or extraordinary result. Courts regularly award lost profits once a patentee has satisfied the four *Panduit* factors without requiring a separate apportionment step or satisfaction of the EMVR. *See, e.g.*, *Presidio Components, Inc. v. Am. Technical Ceramics Corp.*, 702 F.3d 1351, 1359 (Fed. Cir. 2012) (affirming lost profits awarded pursuant to the *Panduit* test without requiring apportionment or satisfaction of the EMVR); *Tate Access Floors, Inc. v. Maxcess Technologies, Inc.*, 222 F.3d 958, 971 (Fed. Cir. 2000) (same). And one district court, addressing apportionment of lost profits directly, explained, "[T]he doctrine of apportionment is inapplicable in assessing damages once the quantity of sales lost (or of sales at eroded prices) has been determined." *W.L. Gore & Assocs., Inc. v. Carlisle Corp.*, No. CIV.A. 4160, 1978 WL 21430, at *11 (D. Del. May 17, 1978).

This is true even where the amount of profits the patentee lost exceeds the amount of profits the infringer gained from its infringing sales. *See* Mark A. Lemley, *Distinguishing Lost Profits from Reasonable Royalties*, 51 Wm. & Mary L. Rev. 655, 669 (2009) ("To make those patent owners whole, defendants must be made to pay in many cases more than they made by infringing ....").[4] As this Court explained in *Pall Corp. v. Micron Separations, Inc.*, "an infringer's sales at a lower price do not defeat the patentee's recovery of its losses at the patentee's price, for the principle of patent damages is to return the patentee to the pecuniary position it would have been in but for the infringement." 66 F.3d 1211, 1223 (Fed. Cir. 1995) (citing *Aro Mfg.*, 377 U.S. at 507; *Rite-Hite*, 56 F.3d at 1544).

B. This Court's En Banc Opinion In *Rite-Hite v. Kelley* Dictates That Lost Profits Should Not Be Apportioned

This Court's en banc opinion in *Rite-Hite v. Kelley* establishes that, where "but for" causation is shown, lost profits should not be apportioned. 56 F.3d at 1548. In *Rite-Hite*, the patent owner sought to obtain lost profits from lost sales of a product that competed with the infringing device but did not practice the asserted patent. *Id.* at 1544. The district court awarded lost profits and the infringer appealed. *Id.* The infringer argued that "to recover damages in the form of lost profits a patentee must prove that, 'but for' the infringement, it would have sold a product covered by the patent in suit ...." *Id.* This Court rejected the infringer's

[4] http://scholarship.law.wm.edu/wmlr/vol51/iss2/10.

position, finding that the patentee was entitled to lost profits for lost sales resulting from the infringement, regardless of whether the lost sales were of products that practiced the patent. *Id.*

Rite-Hite rejects the notion that lost profits must be apportioned absent satisfaction of the EMVR for at least three reasons. First, *Rite-Hite* makes clear that the value of the patent is considered as part of the "but for" analysis, not in a separate apportionment step. Second, *Rite-Hite* affirms a lost profits award for which the type of apportionment advocated by Synopsys is impossible. Third, *Rite-Hite* establishes that the EMVR and the *Panduit* test are separate, independent tests for determining lost profits that are not used concurrently.

1. The Value Of The Patent Is Considered As Part Of The "But For" Analysis

Rite-Hite makes clear that the value of the patent is considered as part of the "but for" analysis. Responding to the infringer's argument that "the intrinsic value of the patent in suit is the only proper basis for a lost profits award," the Court explained:

> [The infringer's] concern that lost profits must relate to the "intrinsic value of the patent" ***is subsumed in the "but for" analysis***; if the patent infringement had nothing to do with the lost sales, "but for" causation would not have been proven.

Rite-Hite, 56 F.3d at 1548 (emphasis added). To demonstrate "but for" causation, a patentee must show that, absent infringement, the patentee would have made

sales that were made by the infringer. *Id.* at 1545. If the asserted patent is not valuable to consumers, the patentee will not be able to meet this burden.

When a patentee demonstrates "but for" causation using the *Panduit* test, as Mentor did in this case, the intrinsic value of the patent is weighed in the analysis of the second *Panduit* factor, the "absence of acceptable non-infringing substitutes." *See, e.g.*, *Depuy Spine, Inc. v. Medtronic Sofamor Danek, Inc.*, 567 F.3d 1314, 1329 (Fed. Cir. 2009). If the patented feature is critical to sales of the infringing product, there *will be no* acceptable non-infringing substitutes, favoring a finding of "but for" causation and an award of lost profits. *Id.* at 1330-31. If the patented feature is not important to sales of the infringing product, there *will be* acceptable non-infringing substitutes, making it impossible for the patentee to demonstrate "but for" causation and precluding an award of lost profits. *Id.* And if the patented feature falls somewhere between—e.g., it is critical to some consumers and unimportant to others—the patentee will be able to show "but for" causation for only some of the sales made by the infringer and will be entitled to lost profits for only those sales. *Id.* It is not necessary or appropriate to further reduce the patentee's recovery by apportioning those lost profits.

In this case, the district court's jury instruction required the jury to consider every non-infringing substitute proposed by Synopsys. A164-71. In particular, the instruction stated that, once Mentor had made a prima facie showing of "but for"

causation, "the burden shift[ed] to Synopsys to show that Mentor would not have made some or all of the diverted sales 'but for' the infringement." A164-65. "Synopsys may do so by showing, for example, any of the following":

> 1. That the Cadence emulation system was an acceptable, available, non-infringing alternative … ;
> 2. That an FPGA prototype was an acceptable, available, non-infringing alternative … ;
> 3. That a software simulator was an acceptable, available, non-infringing alternative … ;
> 4. That Synopsys could have made available during the damages period an acceptable, non-infringing alternative … ; and
> 5. That Intel would have bought fewer or no emulation systems in place of those it bought from Synopsys.

Id. The jury expressly considered these proposed substitutes and still found that, absent Synopsys' infringement, Mentor would have made a majority of the sales that Synopsys made during the infringement period. A182-87. In making this determination, the jury necessarily concluded that the patented feature was ***so critical*** to the majority of Synopsys' sales that none of these non-infringing alternatives would have sufficed.

That there are other features that drive demand for the infringing ZeBu emulators does not change this result. Proof of "but for" causation requires proof that, absent infringement, the patentee would have made sales that were made by the infringer. *Rite-Hite*, 56 F.3d at 1545. The patentee need not prove that the infringing feature is the only reason consumers purchased the infringing product,

only that consumers would have purchased the patentee's product if the infringing product lacked the infringing feature. *Id.* Abundant evidence was presented at trial that consumers would not have purchased Synopsys' ZeBu emulators without the infringing flexible and value change probes. *See supra* § II(C), pp. 6-8.

2. Lost Profits From Lost Sales Of Products That Do Not Practice The Patent Cannot Be Apportioned Based On The Patented Feature

This Court's holding in *Rite-Hite* that a patentee may obtain lost profits for lost sales of products that do not practice the patent is also directly at odds with Synopsys' position that lost profits must be apportioned. Lost profits reflect the profits a patentee would have made absent infringement and are measured based on the cost and selling price of the patentee's product, *not* on the cost and selling price of the infringer's product. Where a patentee seeks compensation for lost sales of products (multicomponent or otherwise) that do not practice the patent, the apportionment urged by Synopsys as necessary is not even possible. Profits from sales of products that do not practice the patent cannot be "apportioned" to reflect the value of the patented feature because ***they do not contain the patented feature***. Through the *Rite-Hite* holding, this Court flatly rejected Synopsys' assertion that apportionment of lost profits is required.

3. EMVR And *Panduit* Are Separate Tests

Rite-Hite makes clear that the EMVR is not applied concurrently with the *Panduit* factors, but is instead an alternative and wholly separate test for determining lost profits. 56 F.3d at 1548-49. In *Rite-Hite*, the court awarded lost profits for lost sales of the patentee's products based on the patentee's proof of "but for" causation using the *Panduit* test. *Id.* Then, in a separate analysis that did not include consideration of the *Panduit* factors, the court applied the EMVR to determine whether the patentee was also entitled to lost profits on convoyed sales of separate products. *Id.* at 1549-50; *see also DePuy Spine*, 567 F.3d at 1331 (distinguishing a *Panduit* lost profits analysis from an EMVR lost profits analysis). The Court never applied the EMVR and the *Panduit* factors ***together*** to determine the amount of lost profits to be awarded.

That *Rite-Hite* rejects the application of the EMVR in the *Panduit* context is confirmed by the opinion offered by Judge Nies in dissent. *Rite-Hite*, 56 F.3d at 1556 (Nies, J., dissenting). Judge Nies criticized the majority for "eviscerat[ing]" any requirement that "lost profits awards [be] dependent, inter alia, on proof that consumer demand for the patentee's good is created by the advantages of the patented invention," *i.e.*, on satisfaction of the EMVR. *Id.* at 1569. This dissent makes clear that, where "but for" causation is proven using the *Panduit* factors, the EMVR is inapplicable.

C. The Cases Relied On By Synopsys Do Not Require Apportionment Of Lost Profits

Synopsys relies on three type of cases to argue that apportionment of lost profits is required: (1) cases regarding apportionment of royalties, (2) cases regarding apportionment of an infringer's disgorged profits, and (3) cases analyzing whether "but for" causation has been proven at all. None of these cases support Synopsys' position.

1. Reasonable Royalty and Disgorgement Cases

Synopsys improperly relies on several opinions regarding apportionment of reasonable royalties and disgorged profits to argue that apportionment of lost profits is required. *See* SPB 44-45, 52-53[5]; *see also* SPB 44-45, 47, 50.[6] Cases apportioning reasonable royalties and disgorged profits do not dictate that lost profits must also be apportioned.

[5] Citing *Ericsson, Inc. v. D-Link Sys., Inc.*, 773 F.3d 1201, 1232 (Fed. Cir. 2014) (apportioning royalties); *VirnetX, Inc. v. Cisco Sys., Inc.*, 767 F.3d 1308, 1329 (Fed. Cir. 2014) (same); *LaserDynamics, Inc. v. Quanta Computer, USA, Inc.*, 694 F.3d 51, 67 (Fed. Cir. 2012) (same); *ResQNet.com, Inc. v. Lansa, Inc.*, 594 F.3d 860, 870 (Fed. Cir. 2010) (same); *Lucent Technologies, Inc. v. Gateway, Inc.*, 580 F.3d 1301, 1324 (Fed. Cir. 2009) (same).

[6] Citing *Dowagiac Mfg. Co. v. Minn. Moline Plow Co.*, 235 U.S. 641, 646 (1915) (apportioning disgorged profits); *Westinghouse Elec. & Mfg. Co. v. Wagner Elec. & Mfg. Co.*, 225 U.S. 604, 614-15 (1912) (same); *Seymour v. McCormick*, 57 U.S. 480, 491 (1853) (same); *Garretson v. Clark*, 111 U.S. 120, 121 (1884) (same).

a. Apportionment Of Reasonable Royalties Does Not Require Apportionment Of Lost Profits

Reasonable royalties and lost profits are different measures of damages awarded pursuant to different standards. A reasonable royalty is intended to compensate a patentee for lost licensing fees, and reflects the amount an infringer would have agreed to pay to license the patent in a hypothetical negotiation taking place before the infringement commenced. *Lucent*, 580 F.3d at 1324-25. Lost profits, on the other hand, are intended to compensate the patentee for sales the patentee would have made absent infringement. *Rite-Hite*, 56 F.3d at 1545.

It is logical to require apportionment absent satisfaction of the EMVR in the reasonable royalty context. If the parties had negotiated a license before the infringement commenced, the infringer would not have agreed to pay more for the license than the infringer's perceived value of the patented feature. Indeed, the *Georgia-Pacific* factors—often weighed in determining the appropriate amount of a reasonable royalty, but not applicable to a determination of lost profits—specifically address apportionment issues, requiring consideration of "the portion of profit or selling price customary … to allow for the use of the invention" (factor 12) and "the portion of realizable profit attributable to the invention as distinguished from non-patented elements [and] significant features or improvements added by the infringer" (factor 13). *Lucent*, 580 F.3d at 1332, 1335.

b. Apportionment Of Disgorged Profits Does Not Require Apportionment Of Lost Profits

A defendant's disgorged profits and a plaintiff's lost profits are also different measures of damages awarded pursuant to different standards. Indeed, disgorged profits have not been available as a remedy for patent infringement since 1946 when Congress amended the patent statute. *See Gen. Motors*, 461 U.S. at 654-55. Disgorged profits were intended to allow a patentee to recover "the fruits of the infringement even if it caused [the patentee] no injury." *Id.* This is directly contrary to a lost profits award, which is based exclusively on the patentee's injury, *i.e.*, "how much [the patentee] would have made if [the infringer] had not infringed." *Aro Mfg.*, 377 U.S. at 507. Because disgorged profits are based not on the patentee's injury but on the "fruits of the infringement," it is logical that such profits should reflect only the portion of the defendant's profits derived from the defendant's unauthorized use of the patented technology.

Notably, the statement from this Court's opinion in *Ericsson*, relied on heavily by Synopsys, that "apportionment is required even for non-royalty forms of damages," cites a single case, *Garretson v. Clark*, in which a defendant's disgorged profits, not a plaintiff's lost profits, were apportioned. *Ericsson*, 773 F.3d at 1226 (citing *Garretson*, 111 U.S. at 121). While it is true that both royalties and a defendant's disgorged profits must be apportioned, neither this

Court nor the Supreme Court has ever held that a patentee is only entitled to a portion of the profits it lost as a result of the infringement.

c. Neither Synopsys Nor The Amici Provide Any Workable Proposal For Apportioning Lost Profits

Synopsys and the amici raise the apportionment analyses set forth in reasonable royalty and disgorgement cases, arguing they are equally applicable in the lost profits context, but provide no practical guidance on how courts may accomplish the proposed apportionment. Nor can Mentor imagine a method by which lost profits could be effectively apportioned. Reasonable royalties and disgorged profits are determined based on ***the infringer's profits***, some portion of which is necessarily derived from the patented feature. Lost profits, on the other hand, are based on ***the patentee's profits***. It is nonsensical, if not impossible, *see supra* § II(B)(2), pp. 35-36, to require apportionment of the patentee's profits based on the technical composition of the infringer's product.

2. Cases Regarding Proof Of "But For" Causation

Synopsys improperly relies on certain post-*Rite-Hite* opinions issued by this Court regarding inadequate proof of "but for" causation to argue that apportionment of lost profits is required. *See* SPB 51-52 (citing *Calico Brand, Inc. v. Ameritex Imports, Inc.*, 527 F. App'x 987, 995-96 (Fed. Cir. 2013); *Ferguson Beauregard/Logic Controls v. Mega Sys., LLC*, 350 F.3d 1327, 1346 (Fed. Cir. 2003)). To the extent these opinions are inconsistent with *Rite-Hite*, *Rite-Hite*, the

en banc, precedential opinion, governs. Regardless, these cases are not inconsistent with *Rite-Hite*. Contrary to Synopsys' assertions, neither case stands for the proposition that, in addition to proving "but for" causation, patentees must apportion or satisfy the EMVR before obtaining lost profits. *See Calico*, 527 F. App'x at 995-96; *Ferguson Beauregard*, 350 F.3d at 1346.

In *Calico*, this Court did not address apportionment or application of the EMVR, but rather analyzed whether "but for" causation had been proven at all. 527 F. App'x at 997. The product at issue in *Calico* was a simple utility lighter. *Id.* at 989. Finding that the defendant sold both infringing and non-infringing lighters, and that the two were completely interchangeable, this Court concluded that, in the absence of the infringing lighter, the defendant would simply have sold the non-infringing lighter. *Id.* at 997. The plaintiff thus would not have made any additional sales "but for" the defendant's infringement. *Id.*

Similarly, in *Ferguson-Beauregard*, this Court remanded for a determination of which of the patentee's products would have been sold "but for" the infringement, not for the type of apportionment urged by Synopsys. 350 F.3d at 1345-46. Ferguson sold two different products that competed with the infringing device: the AutoCycle device and the LiquidLift device. *Id.* The AutoCycle device was a more complex and presumably more expensive device that embodied the inventions claimed in two asserted patents, the '376 Patent and the '991 Patent.

Id. The LiquidLift device, on the other hand, was a less complex and presumably less expensive device that embodied the invention claimed in only one asserted patent, the '376 Patent. *Id.* After a bench trial, the district court determined that the defendant's "Mega Systems' APC 1000 controller did not infringe either the '721 Patent or '991 Patents[, but did] infringe the '376 Patent." *Ferguson-Beauregard v. Mega Sys. LLC*, No. Civ. A. 6:99CV437, 2001 WL 34771614, at *70 (E.D. Tex. Dec. 13, 2001). The district court then awarded lost profits based exclusively on sales of the more expensive, more complex AutoCycle device. *Id.*

This Court reversed the lost profits award on the basis that Ferguson had not demonstrated that "but for" infringement of the '376 Patent it would have made all sales of its more complex, more expensive AutoCycle device. *Ferguson-Beauregard*, 350 F.3d at 1345-46. Because the less expensive, less complex LiquidLift device also competed with the infringing device, it was possible that "but for" the infringement Ferguson would have made some sales of the AutoCycle device and some sales of the LiquidLift Device, or even all sales of the LiquidLift device. This Court remanded for a determination of which sales—sales of the Autocycle device, the LiquidLift device, or both—would have been made "but for" the defendant's infringement of the '376 Patent. *Id.* Nowhere in *Ferguson-Beauregard* did this Court or the lower court suggest that Ferguson was

entitled to only a portion of the profits it lost as a result of the defendant's infringement.

D. Failure To Apportion Lost Profits Will Not Result In Multiple Recoveries

Failure to apportion lost profits will not, as the amici argue, "allow multiple entities to obtain lost profits on the same product where each entity holds a patent on different 'but for' features of the same product." AB 11. The established lost profits analysis protects against multiple recoveries of the same lost profits by requiring consideration of a "but for" marketplace from which only the defendant's infringing product is excluded. *Grain Processing Corp. v. Am. Maize-Prods. Co.*, 185 F.3d 1341, 1350 (Fed. Cir. 1999). Competitors' products are expressly considered, and the patentee may only recover profits for sales the patentee would have made, not sales its competitors would have made. *Id.* In this case, the jury was asked to determine whether Cadence, the only alleged competitor of Mentor and Synopsys for sales to Intel, would have made some of the sales Synopsys made to Intel absent the infringement. A164-71. And in the multi-competitor emulation market, Mentor only sought to recover sales proportional to its market share, accounting for the sales that Mentor's competitors would have made. *Id.* If Cadence were to sue in a subsequent action, Synopsys would be free to rely on the same market evidence and the jury verdict in this case to argue that, in a market

without Synopsys, Mentor, not Cadence or some other competitor, would have made at least some of the sales that were made by Synopsys.

Accordingly, this Court should affirm the jury's award of lost profits.

III. THE DISTRICT COURT PROPERLY APPLIED ASSIGNOR ESTOPPEL TO BAR SYNOPSYS FROM CHALLENGING THE VALIDITY OF THE '376 PATENT

Neither *Lear* nor the facts of this case provide a reason for the en banc Court to revisit the assignor estoppel doctrine. As this Court has concluded, *Lear*, which eliminated the doctrine of licensee estoppel, spoke of assignor estoppel only in dicta, and the policies behind the two doctrines are different. *Diamond Scientific*, 848 F.2d at 1223. Among other differences, "[u]nlike the licensee, who, without *Lear* might be forced to continue to pay for a potentially invalid patent, the assignor who would challenge the patent has already been fully paid for the patent rights." *Id.* at 1224.

Application of assignor estoppel is particularly appropriate here. While Dr. Burgun was employed by Mentor from 1996 until 2000, he enjoyed Mentor's support in the form of salary, benefits, and resources as Mentor's head of research and development. A1994(20:7-18). In 1998, when Burgun assigned his inventions to the company, he agreed to "generally assist [Mentor] in securing and maintaining proper patent protection" for the assigned inventions, and following the assignment he executed at least one additional document in connection with the

application for the '376 Patent. A1571; A1573-74; A1579. Burgun left Mentor to form EVE, where he acted as President and CEO, owned stock, and was involved in the technical development of EVE's products, including ZeBu emulators. A1619; A1620; A1627; A1650; A1656. After Mentor sued EVE for infringement, Burgun helped obtain a license from Mentor that enabled EVE to continue to make the infringing ZeBu systems. A1982(¶ 1); A1984. Burgun then facilitated Synopsys' acquisition of EVE despite knowing that it would terminate the Mentor-EVE License. A1984; A2518. Synopsys made Burgun Vice President of its Emulation Team, issued him Synopsys stock, and continued to manufacture and sell ZeBu emulators. A1761; A1936; A2035; A2611-12. At the time of trial, Burgun was still a part-time employee of Synopsys. A41596:2-4. Burgun's significant role in Synopsys' infringement belies Synopsys' suggestion that the district court applied assignor estoppel simply because it "briefly employed Burgun." SPB 42.

Synopsys' relationship with Burgun was unquestionably "close[]" "in light of the act of infringement." *Shamrock Tech., Inc. v. Med. Sterilization, Inc.*, 903 F.2d 789, 793 (Fed. Cir. 1990). This case fits the classic privity scenario that estops "[the] corporation founded by the assignor" (EVE) from challenging the validity of the assigned patents. *See Diamond Scientific*, 848 F.2d at 1224. And because of Burgun's past work at EVE developing the ZeBu Server, which he then

sold to Synopsys, and his continued employment at Synopsys, Synopsys "availed itself of [Burgun's] 'knowledge and assistance' to conduct infringement" of a patent Burgun had previously assigned to Mentor. *See Intel Corp. v. Int'l Trade Comm'n*, 946 F.2d 821, 839 (Fed. Cir. 1991) (quoting *Shamrock*, 903 F.2d at 794). As a result, "the balance of equities between the parties" requires that Burgun's privy, Synopsys, not be permitted to "destroy" the rights Burgun assigned to his invention by "derogating the patent['s] validity." *See Diamond Scientific*, 848 F.2d at 1226.

Accordingly, this Court should affirm the district court's summary judgment order estopping Synopsys from challenging the validity of the '376 Patent.

IV. THE DISTRICT COURT CORRECTLY HELD SYNOPSYS' '109 PATENT INDEFINITE

The sole independent claim of the '109 Patent requires "displaying" circuit information "near" corresponding portions of the source code. A500(22:52-56). The district court correctly held that the '109 Patent is indefinite because the claims, read in light of the specification and prosecution history, fail to inform, with reasonable certainty, one of skill in the art what is or is not "near." *Nautilus, Inc. v. Biosig Instruments, Inc.*, 134 S. Ct. 2120, 2128 (2014).

A. The Patent And File History Fail To Inform With Reasonable Certainty The Scope Of The Term "Near"

The claims of the '109 Patent do not define the requisite degree of "near." *See* A14228, A14305. Without guidance from the claims, the district court was tasked with determining "whether the patent's specification provides some standard for measuring that degree." *Datamize, LLC v. Plumtree Software, Inc.*, 417 F.3d 1342, 1351 (Fed. Cir. 2005). The district court correctly held that it does not.

The term "near" appears in the specification eight times. *See* A455(Abstract); A493(7:58, 8:9, 8:37, 8:60); A496(14:22); A500(22:54). But none of those eight instances define "near," describe examples of what constitutes "near," or otherwise provide any standard for measuring "near."

Synopsys selectively points to two figures that purportedly show examples of "near." The specification, however, never uses the term "near" when describing any figures. And Synopsys conveniently ignores Figure 30, which shows circuit information (3030) and related source code (3020) on almost opposite corners of the screen:

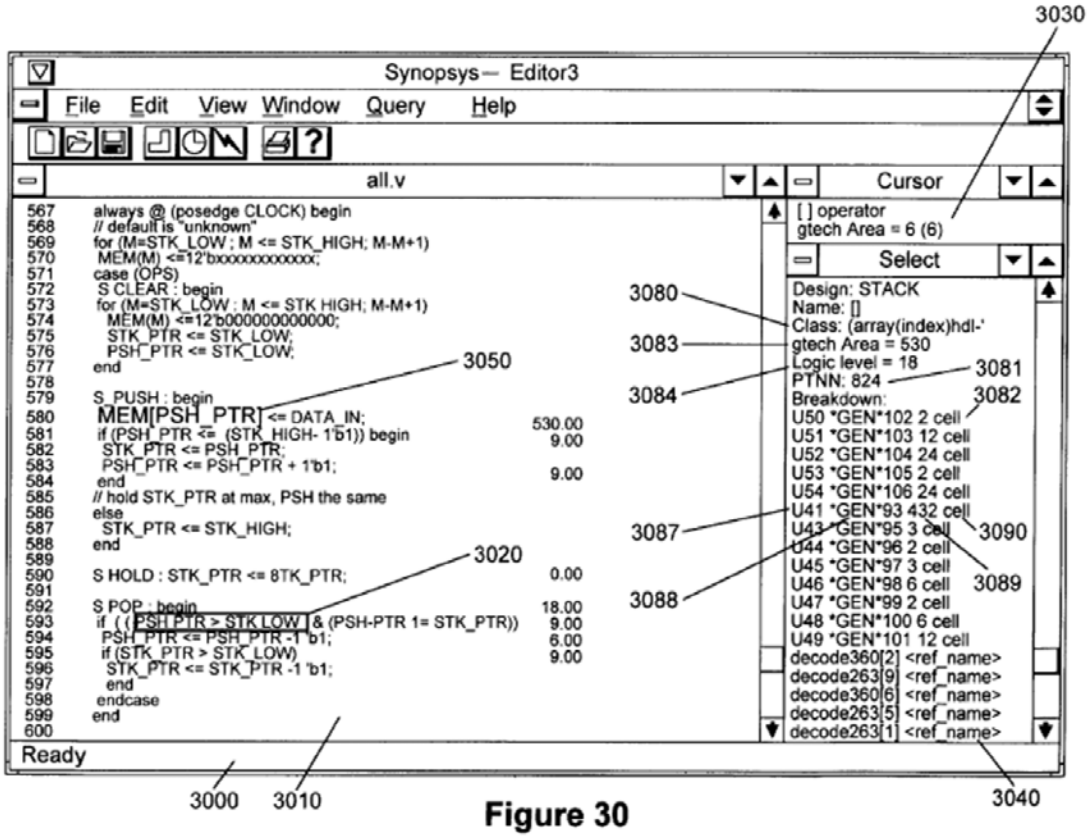


Figure 30

A486; A494(10:44-45). The stark contrast between this figure and the figures presented by Synopsys, in which the items are adjacent, illustrates the ambiguity of the term "near." "Such ambiguity falls within 'the innovation-discouraging 'zone of uncertainty' against which [the Supreme Court] has warned.'" *Interval Licensing LLC v. AOL, Inc.*, 766 F.3d 1364, 1374 (Fed. Cir. 2014) (quoting *Nautilus*, 134 S. Ct. at 2130).

Nor does the prosecution history provide any guidance for the meaning of "near." The Applicant referred to the "displaying" limitation twice during prosecution but never discussed the meaning of "near." *See* A15325, A15329.

B. Synopsys' Interpretation Of "Near" Fails To Provide Reasonable Certainty Because It Is Subjective

Synopsys proposes that "near" means "close enough to enable the engineer to make the mental connection between the source code and the reported test result." SPB 62-63. The mere fact that Synopsys can articulate a definition does not save "near" from being indefinite "if a person of ordinary skill in the art cannot translate the definition into meaningfully precise claim scope." *See Halliburton Energy Servs., Inc. v. M-I LLC*, 514 F.3d 1244, 1251 (Fed. Cir. 2008). "Although absolute or mathematical precision is not required, it is not enough … to identify '*some standard* for measuring the scope of the phrase.'" *Interval Licensing*, 766 F.3d at 1370-71. Rather, the context of the invention "must provide objective boundaries for those of skill in the art." *Id.* at 1371.

Terms of degree, such as "near," are not meaningfully precise if they depend on subjective opinions. In *Interval Licensing*, this Court held that displaying content on a device in an "unobtrusive manner" was indefinite because the specification failed to provide "objective boundaries" for the limitation. 766 F.3d at 1371. "[A] term of degree fails to provide sufficient notice of its scope if it depends 'on the unpredictable vagaries of any one person's opinion.'" *Id.* (quoting *Datamize*, 417 F.3d at 1350). Similarly, in *Datamize*, this Court held that "aesthetically pleasing" was indefinite because it "depend[s] solely on the

unrestrained, subjective opinion" of individuals practicing the invention. 417 F.3d at 1350.

Synopsys' definition is entirely subjective and contains no "objective anchor." *See id.* Like displaying in an "unobtrusive manner," different engineers would "make the mental connection" at varying degrees of "nearness" depending on their experience, visual acuity, familiarity with the software, or any number of "unpredictable vagaries." *See Interval Licensing*, 766 F.3d at 1371. Such reliance on the "unrestrained, subjective opinion" of the person practicing the invention confirms that the specification fails to disclose with reasonable certainty the scope of the term "near."

C. Synopsys' Inconsistent Position On Separate Windows Demonstrates The Uncertainty Of The Term "near"

Synopsys has been inconsistent about whether items in separate windows can be "near" each other. In its preliminary infringement contentions, Synopsys asserted that Mentor's Veloce met the "near" limitation by displaying information in separate, but adjacent windows. A14986.

Synopsys switched positions when Mentor cited separate but adjacent windows in prior art. *See* A14307-08. Synopsys' expert, Dr. Hutchings, opined in his validity report that a prior art reference did not meet the "near" limitation because the "circuit information and the source text are displayed in different and separate windows," A14872, and reasserted in his deposition that items in separate

windows could not be "near." A14943. Synopsys' assertion that two items could be "near" in the same window but "not near" even if they were the same distance apart but in separate windows demonstrates the ambiguity of "near."

D. *Young v. Lumenis* Is Inapposite

Synopsys' reliance on *Young v. Lumenis, Inc.*, 492 F.3d 1336 (Fed. Cir. 2007), is misplaced. First, *Young* applied the "amenable to construction" and "insolubly ambiguous" standard that was overruled and replaced with the "reasonable certainty" standard by the Supreme Court in *Nautilus*.

Second, *Young*'s claimed surgical technique for removing a cat claw by making an incision "near the edge of the ungual crest" is factually distinguishable. *Young*, 492 F.3d at 1340. The specification explained that the incision "severs the most distal portion of the epidermis from the underlying fascia of the ungual crest" and "is made near the most distal edge of the epidermis and extends circumferentially around the claw to sever the epidermis from the ungual crest." *Id.* at 1346. These descriptions provided objective limitations based on the medical implications of the incision and the physical characteristics of a cat claw.

There are no such descriptions or limitations here. The '109 Patent is completely devoid of any discussion of the term "near," and the only definition that Synopsys can devise has no objective boundaries to limit the scope of the term.

E. There Are No Genuine Issues Of Material Fact

Indefiniteness is a question of law, and the competing opinions of Dr. Hutchings and Dr. Sarrafzadeh do not create a genuine issue of material fact. A court does not make factual findings when it considers expert testimony regarding indefiniteness. *See Exxon Research & Eng'g Co. v. United States*, 265 F.3d 1371, 1376 (Fed. Cir. 2001). Dr. Hutchings' conclusory opinions do not create triable issues of fact. *See Sitrick v. Dreamworks, LLC*, 516 F.3d 993, 1001 (Fed. Cir. 2008); *Techsearch, LLC v. Intel Corp.*, 286 F.3d 1360, 1372 (Fed. Cir. 2002). The only facts relevant to this indefiniteness inquiry are undisputed and contained in the '109 Patent disclosure. Thus, summary judgment that the '109 Patent is indefinite was appropriate.

V. THE DISTRICT COURT CORRECTLY HELD SYNOPSYS' '526 PATENT INELIGIBLE

The district court correctly held that Claims 19, 24, 28, 30, and 33 of the '526 Patent are directed to patent-ineligible subject matter under 35 U.S.C. § 101. These claims recite a "machine-readable medium," which the specification defines to include "carrier waves." A627(52:31-36); A627-29(52:51-58:3). Because carrier waves are unpatentable, *In re Nuijten*, 500 F.3d at 1354, the district court's judgment should be affirmed.

In *In re Nuijten*, this Court held that transitory signals, such as carrier waves, are unpatentable because they do not fall under any of the four statutory categories

of patentable subject matter. 500 F.3d at 1357. The Court analyzed whether claims related to a signal processing technique could be patentable if they claimed the "signals themselves." *Id.* at 1351. Noting that related claims directed to "a method of embedding … a signal," "encoder means for encoding the signal," and a "storage medium having stored thereon a signal" were patentable, the court held that, in contrast, claims directed to the "signals themselves" did not qualify as a "process, machine, manufacture, or composition of matter." *Id.* at 1348, 1351-52, 1357.

The claims of the '526 Patent are likewise fatally directed to the "signals themselves." The specification defines "machine-readable medium" to include carrier waves. A627(52:31-36). It is well established that "claims must be construed so as to be consistent with the specification." *Merck & Co. v. Teva Pharm. USA, Inc.*, 347 F.3d 1367, 1371 (Fed. Cir. 2003). The axiom that claims should be construed to preserve validity does not apply where, as here, the specification provides an unambiguous definition. *Liebel-Flarsheim Co. v. Medrad, Inc.*, 358 F.3d 898, 911 (Fed. Cir. 2004).[7]

[7] Synopsys' pronouncement that the asserted claims are *Beauregard* claims is immaterial. Like any other claim, a *Beauregard* claim is invalid if it is directed to patent-ineligible material. *See, e.g.*, *Alice Corp. Pty. v. CLS Bank Int'l*, 134 S. Ct. 2347, 2360 (2014); *CyberSource Corp. v. Retail Decisions, Inc.*, 654 F.3d 1366, 1376 (Fed. Cir. 2011).

Synopsys could have drafted the claims, amended the claims during prosecution, or reissued the '526 Patent before filing suit against Mentor to limit the claims to non-transitory subject matter, but failed to do so. A related patent owned by Synopsys, U.S. Patent No. 8,099,271—which was filed only one day after the '526 Patent, has a virtually identical specification, lists the same inventors, and claims priority to the same two provisional applications—limits its "machine-readable medium" claims by using the term "non-transitory." A23485-86(52:33-53:44).

The statutory embodiments of "machine-readable medium" do not prevent the claims from covering patent-ineligible subject matter. The Patent Office addresses the exact situation presented here:

> A claim that covers both statutory and non-statutory embodiments … embraces subject matter that is not eligible for patent protection and therefore is directed to non-statutory subject matter …. Thus, *a claim to a computer readable medium that can be a compact disc or a carrier wave covers a non-statutory embodiment and therefore should be rejected under 35 U.S.C. 101*.

MPEP 2106 (9th ed. Mar. 2014) (emphasis added). The Patent Office consistently rejects claims directed to machine- or computer-readable media that include transitory signals, even if statutory subject matter is also included.[8]

[8] *E.g.*, *Ex Parte David Robert Draeger, et al.*, Appeal 2009-011120, 2011 WL 3202799, at *4 (Bd. Pat. App. & Interf. July 21, 2011) (rejecting "computer readable medium" claims under § 101 for embracing carrier waves and statutory subject matter); *Ex Parte Michael Scott Montgomery, et al.*, Appeal 2010-004770,

Nor do references to hardware render the asserted claims patentable. In *Digitech Image Technologies, LLC v. Electronics for Imaging, Inc.*, this Court rejected the argument that claims directed to unpatentable subject matter were converted to eligible status by operating with patentable hardware. 758 F.3d 1344, 1350 (Fed. Cir. 2014). A claim at issue read: "A device profile for describing properties of a device in a digital image reproduction system to capture, transform or render an image." *Id.* at 1349. The "device profile" was held to be unpatentable "ethereal, non-physical" information. *Id.* at 1350. And the device profile's association with "a device in a digital image reproduction system," i.e., patentable hardware such as "digital cameras, monitors, TVs, printers, etc.," did not change the fact that the claims were directed to ineligible subject matter. *Id.* at 1347, 1349.

Likewise, references to hardware in the claims of the '526 Patent do not change the fact that the claims cover unpatentable carrier waves. And the claims here are not limited to "specialized hardware," as Synopsys asserts. The "machine-readable medium" contains instructions that can be executed on a "data processing system," which in turn debugs a separate "electronic system" that has

---

2012 WL 3863260, at *2 (Bd. Pat. App. & Interf. Aug. 31, 2012) (same); *Ex Parte Elliot Schwartz*, Appeal 2009-006734, 2011 WL 6281796, at *1-2 (Bd. Pat. App. & Interf. Dec. 12, 2011) (same); *Ex Parte Brian D. Gragg*, Appeal 2009-001469, 2009 WL 3183254, at *3 (Bd. Pat. App. & Interf. Oct. 5, 2009) (same).

"instrumentation circuitry included therein." A629(55:18-21). The "machine-readable medium" is the subject matter of the claims, not the data processing or electronic system or any other "specialized hardware" alleged by Synopsys.

The Court should affirm the judgment that the '526 Patent is patent-ineligible.

ARGUMENT IN FAVOR OF MENTOR'S CROSS-APPEAL

VI. THE DISTRICT COURT ERRED IN BARRING MENTOR'S WILLFULNESS CLAIMS

A. Standard of Review

The district court disposed of Mentor's willfulness claim on the objective prong by granting Synopsys' motion *in limine* shortly before trial. A26720; A137. The objective determination of willfulness is a question of law reviewed de novo. *Bard Peripheral Vascular, Inc. v. W.L. Gore & Assocs., Inc.*, 682 F.3d 1003, 1007 (Fed. Cir. 2012).

B. Synopsys And EVE Willfully Disregarded The '376 Patent And The Likelihood Of Infringement

Clear and convincing evidence establishes that Synopsys "acted despite an objectively high likelihood that its actions constituted infringement of a valid patent," thus satisfying the objective prong of willfulness. *In re Seagate Tech., LLC*, 497 F.3d 1360, 1371 (Fed. Cir. 2007) (en banc).[9] EVE was aware of the '376

[9] Mentor preserves the issue of whether willfulness should be provable by a preponderance of the evidence, consistent with *Octane Fitness, LLC v. ICON*

Patent and knew its license to practice the patent would terminate if it were acquired by a Mentor competitor. A1984. Synopsys targeted EVE for acquisition specifically because EVE's ZeBu emulators "fill[ed] that gap" in Synopsys' product line. SPB 12. Synopsys knew that the acquisition would terminate the Mentor-EVE License. *See* A1204(¶¶ 14-15). Synopsys nevertheless acquired EVE and made it a wholly-owned subsidiary. A1849; SBP i. Despite termination of the Mentor-EVE License, Synopsys and EVE continued to make and sell ZeBu emulators. A1926. Synopsys and EVE continued their infringement unabated from the time Synopsys acquired EVE, through the filing of Synopsys' declaratory judgment action and Mentor's willfulness counterclaim, during trial, and after the jury's verdict. *See* A36447-48. Nevertheless, the district court rejected Mentor's willfulness claim before trial.

C. The District Court Failed To Consider The Totality Of Circumstances When It Rejected Mentor's Willfulness Claim Based Solely On The Procedural Posture Of The Case

The district court did not consider the evidence showing that Synopsys and EVE knew they were infringing the '376 Patent, instead rejecting Mentor's willfulness claim "with no analysis of witness demeanor or live testimony or anything else … based just on the procedural posture of the case." A40394:11-13.

Health & Fitness, Inc., 134 S. Ct. 1749, 1758 (2014), which lowered the burden of proof in the "analogous" context of awarding attorney fees. *See Halo Elecs., Inc. v. Pulse Elecs., Inc.*, 780 F.3d 1357, 1361, 1363 (Fed. Cir. 2015) (O'Malley, J., dissenting from denial of rehearing en banc).

The district court's approach was inconsistent with a long line of pre-*Seagate* cases in which this Court established that a determination of willfulness must consider the totality of the circumstances. *See Knorr-Bremse Systeme Fuer Nutzfahrzeuge GmbH v. Dana Corp.*, 383 F.3d 1337, 1342-43 (Fed. Cir. 2004) (en banc), and cases cited therein. *Seagate* did not reject the totality of the circumstances test; in fact, it made clear that the determination of willfulness depends on the facts of each case. *See, e.g.*, *Seagate*, 497 F.3d at 1374 ("in ordinary circumstances, willfulness will depend on an infringer's prelitigation conduct"); *see also id.* ("whether a willfulness claim based on conduct occurring solely after litigation began is sustainable will depend on the facts of each case").

That the "totality of the circumstances" test applies to willfulness also follows from the Supreme Court's recent jurisprudence regarding the "exceptional case" standard for awarding attorney's fees under 35 U.S.C. § 285, which has rejected "overly rigid" formulations and directed district courts to determine exceptionality "in the case-by-case exercise of their discretion, considering the totality of the circumstances." *Octane Fitness*, 134 S. Ct. at 1751. The same should be true for willfulness determinations under Section 284 because this Court's jurisprudence on Sections 284 and 285 have "closely mirrored" each other. *Halo Elecs.*, 780 F.3d at 1361 (O'Malley, J., dissenting from denial of rehearing en banc.)

The district court's rejection of Mentor's willfulness claim based on a rigid interpretation of the procedural posture of the case was inconsistent with these flexible standards.

1. The District Court Failed To Consider Any Conduct By Synopsys Or EVE Prior To Termination Of The Mentor-EVE License

The district court did not consider events prior to Synopsys' filing of a declaratory judgment action, including the history between Burgun, EVE, and Mentor, and Synopsys' plan to continue to sell infringing ZeBu emulators after the EVE acquisition. Indeed, Synopsys' declaratory judgment complaint shows that it expected that the license would terminate and an infringement claim would follow. A1204(¶¶ 14-15). Although infringement by Synopsys and EVE did not begin until the Mentor-EVE license terminated, it does not follow that the court may disregard pre-infringement conduct. In particular, Synopsys' knowledge that EVE was manufacturing and selling ZeBu emulators under a license that would automatically terminate when Synopsys acquired EVE alerted Synopsys to the high likelihood that its continued manufacture and sale of those products would constitute infringement. *See Spindelfabrik Sueseen-Schurr Stahlecker & Grill GmbH v. Schubert & Salzer Maschinenfabrik Aktiengesellschaft*, 829 F.2d 1075, 1084 (Fed. Cir. 1987) (affirming finding of willfulness because patentee's denial of a license made defendant "explicitly aware of the possibility of infringement"); *see*

also Acoustical Design, Inc. v. Control Elecs., Inc., 932 F.2d 939, 940, 942 (Fed. Cir. 1991) (affirming finding of willfulness where infringer "continued to manufacture and sell the same [products] it was previously licensed to manufacture notwithstanding the termination of the agreements").

The failure to consider such facts was error because there is "no precedent for a decision that an infringer must be allowed a certain amount of time to 'develop' willfulness." *Ralston Purina Co. v. Far-Mar-Co*, 772 F.2d 1570, 1577 (Fed. Cir. 1985). "[P]atent infringement is a continuing tort, and an action even if innocently begun does not automatically retain its purity as circumstances change." *Pall Corp.*, 66 F.3d at 1221-22 (affirming finding of willfulness once infringer switched part of its production to non-infringing alternative several years after litigation began).

Consistent with these principles, this Court has considered defendants' conduct before infringement began in determining that they acted willfully. *See, e.g., Metabolite Labs., Inc. v. Lab. Corp. of Am. Holdings*, 370 F.3d 1354, 1370-72 (Fed. Cir. 2004) (affirming willfulness finding that considered actions taken while license was in place); *Bott v. Four Star Corp.*, 807 F.2d 1567, 1572 (Fed. Cir. 1986) (affirming willfulness finding where defendant was aware of plaintiff's design long before the patent issued and knew of patent before it sold an infringing device).

Accordingly, the district court should have considered the conduct of Synopsys and EVE prior to termination of the Mentor-EVE license in assessing willfulness.

2. The District Court Ignored Months Of "Prelitigation" Willful Infringement By Synopsys Between Termination Of The Mentor-EVE License And Mentor's Filing Of A Willfulness Claim

The district court compounded its error by using the wrong date—the date Synopsys filed its declaratory judgment action—as the cutoff between Synopsys' "prelitigation" and "postlitigation" conduct. A40546. *Seagate* explained that the reason to distinguish between "prelitigation" and "postlitigation" conduct is because "when a complaint is filed, a patentee must have a good faith basis for alleging willful infringement," and thus "a willfulness claim asserted in the original complaint must necessarily be grounded exclusively in the accused infringer's pre-filing conduct." *Seagate*, 497 F.3d at 1374.

That logic collapses if the accused infringer's filing of a declaratory judgment action is used as the cutoff between "prelitigation" and "postlitigation" conduct. Synopsys' declaratory judgment complaint (filed September 27, 2012) did not contain a willfulness claim relating to its "pre-filing conduct." A1200-09. No willfulness claim was asserted until Mentor filed its answer and counterclaims on January 11, 2013. A1228(¶ 7). In similar circumstances, courts have recognized the date of the willfulness claim as the only logical cutoff between

"prelitigation" and "postlitigation" conduct. *See Trs. of Univ. of Pa. v. St. Jude's Children's Research Hosp.*, 982 F. Supp. 2d 518, 532 (E.D. Pa. 2013) (holding that the "post-filing period begins at the time a patentee files a willful infringement claim," not when accused infringer files a declaratory judgment action, because "[a] contrary finding would have the bizarre effect of encouraging alleged infringers to file declaratory judgment actions immediately after the issuance of a patent so that they could infringe on valid patents with no fear of a willfulness claim"); *see also Inv. Tech. Grp., Inc. v. Liquidnet Holdings, Inc.*, 759 F. Supp. 2d 387, 412 (S.D.N.Y. 2010), *aff'd sub nom. Liquidnet Holdings, Inc. v. Pulse Trading, Inc.*, 478 F. App'x 671 (Fed. Cir. 2012) (defining filing date as date patent owner filed its willfulness claim, rejecting filing date of subsequent complaint for declaratory relief).

Here, using the correct cutoff date of January 11, 2013 (the date of Mentor's willfulness counterclaim), Synopsys engaged in over three months of "prelitigation" infringement between termination of the Mentor-EVE license and the filing of Mentor's willfulness claim. The district court improperly ignored this months-long, willful infringement.

3. Mentor's Decision Not To Seek A Preliminary Injunction Does Not Bar Its Willfulness Claim

The district court also erred in holding that Mentor's decision not to seek a preliminary injunction categorically disqualified it from arguing that Synopsys'

postlitigation infringement was willful. A40547. *Seagate*'s statement that a patentee that does not seek a preliminary injunction "should not be allowed to accrue enhanced damages based solely on the infringer's post-filing conduct," 497 F.3d at 1374, does not apply here because, again, Mentor's willfulness claim is based on both prelitigation and postlitigation infringement. And, as this Court recently explained, *Seagate* "states no rigid rule" in this regard. *Aqua Shield v. Inter Pool Cover Team*, 774 F.3d 766, 773-74 (Fed. Cir. 2014) (vacating district court's decision of no willful infringement where motion for a preliminary injunction was denied). Just as denial of preliminary injunction is "a legally insufficient reason for determining that [the defendant] did not willfully infringe," *id*., *Seagate* also does not dictate a strict requirement that a patentee must seek a preliminary injunction in all circumstances in order to allege willful infringement.

Indeed, there are many reasons a patentee may decide not to seek a preliminary injunction that are specific to the facts of each case and have nothing to do with whether an infringer is acting willfully. The facts here illustrate that categorical rules are inappropriate. To preserve its claim for willful infringement under the district court's framework, Mentor would have been required to immediately file a motion for preliminary injunction *in the declaratory judgment action filed by Synopsys*, even though (1) Mentor was the defendant; (2) Synopsys waited nearly three months to serve the complaint while the parties discussed

settlement (A917; A31515); (3) Mentor's motion to transfer venue was pending in the court where Synopsys filed its declaratory judgment action for almost three months before it was granted (A918; A924); and (4) Mentor did not yet know the outcome of the IPR of the '376 Patent (A10404; A10455).

There is not—and should not be—a uniform rule requiring a patentee to seek a preliminary injunction to preserve a claim for willfulness.

D. The District Court Erred By Never Considering Whether Synopsys' Defenses Were Reasonable

By rejecting Mentor's willfulness claim based on the "procedural posture of the case," the district court never considered the key issue in determining objective willfulness: whether the infringer's defenses were reasonable. *See Bard Peripheral*, 682 F.3d at 1005-06. Synopsys did not—and cannot—show that its failed defenses were reasonable. Synopsys was barred under assignor estoppel from challenging the validity of the '376 Patent, and, as demonstrated by the single infringement defense it has appealed (*see supra* § I, pp. 14-28), its defenses to infringement were without merit.

Accordingly, this Court should reverse or vacate the district court's willfulness ruling.

VII. THE DISTRICT COURT ERRED IN INVALIDATING CLAIMS 7, 9, AND 13 OF MENTOR'S '882 PATENT FOR FAILURE TO SATISFY THE WRITTEN DESCRIPTION REQUIREMENT

A. Standard Of Review

A district court's grant of summary judgment of invalidity for lack of written description is reviewed de novo. *See Crown Packaging Tech., Inc. v. Ball Metal Beverage Container Corp.*, 635 F.3d 1373, 1380 (Fed. Cir. 2011). The adequacy of the written description is tested by asking whether the disclosure "reasonably conveys to those skilled in the art that the inventor had possession of the claimed subject matter as of the filing date." *Ariad Pharm., Inc. v. Eli Lilly & Co.*, 598 F.3d 1336, 1351 (Fed. Cir. 2010).

A district court should grant summary judgment of invalidity only if the moving party submits such clear and convincing evidence of facts underlying invalidity that no reasonable jury could find otherwise. *SRAM Corp. v. AD-II Eng'g, Inc.*, 465 F.3d 1351, 1357 (Fed. Cir. 2006).

B. Overview Of The '882 Patent

An emulator includes interconnected reconfigurable logic devices, e.g., FPGAs. A565(2:10-11). Each logic device includes logic elements and input/output circuitry. A565(2:11-13). The logic elements are responsible for the logic operations of the device. A566(3:53-58). The input/output circuitry connects the logic devices to each other. *Id.*

Mentor's '882 Patent describes an emulator in which there is at least one "user clock" and at least one "signal routing clock." A565(2:13-19). The user clocks control the timing of logic signals within the logic devices. A566(4:13-16). The signal routing clock controls the timing of the input/output circuitry that routes signals between logic devices. *Id.*

The logic devices have a limited number of input/output pins—far fewer than the number of signals within a logic device—limiting the transfer of signals between logic devices. A6076. One of the inventions disclosed and claimed in the '882 Patent addresses this problem by setting the signal routing clock at a higher frequency than the user clock. A566(4:24-28). This feature, in combination with the use of multiplexers to connect multiple signals to a single pin, provides for high speed transfers of a large number of signals from one logic device to another in time for the signals to be used in the second device. A566(3:63-65; 4:24-30); A6076-77.

If an emulation system includes only one signal routing clock, the clock signal from that clock may need to travel a long distance between the many FPGAs. This distance creates several problems: clock signals are delayed due to increased travel time; and the clock signals arrive at different destinations at different times (known as "skew"). These problems may cause FPGAs to malfunction or behave sub-optimally. These problems may be addressed through

the use of regional signal routing clocks operating independently of the user clocks. These regional clocks greatly reduce the distance over which the signals must travel. As a result, delays and skew are minimized, and FPGAs are more likely to perform as expected. A6078-79.

C. The District Court Construed "independent" To Mean "No Required Timing Relationship Between Clock Edges"

Claim 5, upon which asserted claims 7, 9, and 13 depend, requires that the user clocks and signal routing clocks be "independent." A570(12:35-39). During claim construction, the district court construed the term "independent" to mean "there is no required timing relationship between clock edges." A10848. This construction was proposed by Synopsys, whose expert explained that a "required timing relationship between clock edges" means that the arrival time of a clock edge of one clock has a defined relationship with the arrival time of a clock edge of another clock. *See* A5130.

D. The District Court Erred In Holding Claims 7, 9, And 13 Invalid For Failure To Satisfy The Written Description Requirement

On summary judgment, the district court held that Claim 5 requires "unqualifiedly independent clocking," and found that the '882 Patent fails to demonstrate possession of "unqualifiedly independent clocking" because the specification requires that the signal routing clock have a higher frequency than the user clock. A23749.

This holding is erroneous for three reasons. First, an application for a patent, including the original claims, is a part of the specification and may be relied on to show compliance with the written description requirement. *Ariad*, 598 F.3d at 1349; *In re Gardner*, 480 F.2d 879, 880 (CCPA 1973). Original Claim 1 of the '882 Patent application includes the precise language of Claim 5 that the district court erroneously found to be missing from the written description. A19441. Original Claim 1, a part of the written description, recites input/output circuitry that is "clocked by one or more **signal routing clock signals which are independent of the first and second clock signals**." *Id*. (emphasis added). This language demonstrates that the inventor had possession of "independent" clocks as of the filing date. The meaning of "independent" in original Claim 1 and current Claim 5 must be the same. *See Rexnord Corp. v. Laitram Corp.*, 274 F.3d 1336, 1342 (Fed. Cir. 2001). For that reason, whatever meaning is ascribed to the language of Claim 5, the same language and meaning appeared in the specification as filed and demonstrates possession of the claimed invention at that time.

Second, the requirement that the written description disclose "unqualifiedly independent clocking" is inconsistent with the district court's own claim construction. The construction that "independent" means "there is no required timing relationship between clock edges" requires only independence between clock edges, not between clock frequencies.

Third, that the signal routing clock should have a higher frequency than the user clock does not affect independence because it is not a "required timing relationship between clock edges." Synopsys' expert testified that a timing relationship between clock edges is determined by verifying the relative arrival times of clock edges at their respective components. A5130. The clock edges of the signal routing clock and user clock, however, could arrive at any time relative to each other as long as the signal routing clock has a higher frequency. *See* A14247-48. Mentor's expert Dr. Sarrafzadeh explained that "[t]hose of skill in the art consider characteristics other than frequency if they wish to determine if two clocks are truly independent" and "do not consider [a difference of frequencies] to be a relationship between clock edges." A14248.

The specification teaches that independence is not affected by the frequency relationship between the signal routing clock and the user clock. "Except for the relationship that each of signal routing clock 117 having a higher frequency than an associated user clock 118, signal routing clocks are independent of user clocks." A566(4:16-19). Synopsys' own expert, Dr. Hutchings, cited this passage to support Synopsys' proposed claim construction for "independent," which the district court adopted. A5131-32. When there is such "unambiguous support for [the] claims," invalidity for lack of adequate written description cannot be shown

by clear and convincing evidence. *See Trading Technologies Int'l, Inc. v. Open E Cry, LLC*, 728 F.3d 1309, 1321 (Fed. Cir. 2013).

Yet the district court disregarded the specification, both sides' experts, and its own claim construction, and incorrectly applied a more restrictive interpretation of "independent" by forbidding any relationship at all, even a relationship limited to the relative frequencies of user clocks and signal routing clocks. In doing so, the district court violated the canon of construction that would require the court to apply the construction that makes the claim valid, i.e., its own original claim construction. *Whittaker Corp. by Technibilt Div. v. UNR Indus., Inc.*, 911 F.2d 709, 712 (Fed. Cir. 1990). The district court's summary judgment decision invalidating the '882 Patent should be reversed.

VIII. THE DISTRICT COURT ERRED IN FINDING THAT CERTAIN OF MENTOR'S INFRINGEMENT CLAIMS WERE BARRED BY RES JUDICATA

A. Standard of Review

The district court found on summary judgment that Mentor's claims that Synopsys infringed the '531 and '176 Patents were barred by res judicata. A14061-62. "Whether a cause of action is barred by claim preclusion is a question of law which we review without deference." *Brain Life, LLC v. Elekta Inc.*, 746 F.3d 1045, 1052 (Fed. Cir. 2014).

B. Mentor's Claims Cannot Be Barred Because They Arose After Resolution Of The Prior Action

In the 2006 Action, Mentor asserted that EVE's ZeBu ZV, ZeBu XL, and ZeBu UF emulators (the "old ZeBu emulators") infringed Mentor's '176 and '531 Patents. A1658-64. The 2006 Action was resolved with EVE taking the Mentor-EVE License. A3659-62. After the license terminated on October 4, 2012, when Synopsys acquired EVE, Mentor brought claims alleging that the ZeBu emulators, none of which existed at the time of the 2006 Action, infringed the '176 and '531 Patents. A1216-29. Mentor's claims were based exclusively on acts of infringement that occurred after October 4, 2012. *Id.*

The Supreme Court's holding in *Lawlor* makes clear that a prior judgment on the merits "cannot be given the effect of extinguishing claims which did not even then exist and which could not possibly have been sued upon in the previous case." 349 U.S. at 328. The facts of *Lawlor* are similar to the present case. In *Lawlor*, the petitioners brought an antitrust action against the National Screen Service Corporation, alleging that National Screen "had conspired to establish a monopoly in the distribution of [movie posters] …." *Id.* at 324. The case settled with National Screen agreeing to furnish plaintiffs with movie posters "at specified prices" pursuant to a "sublicense." *Id.* Six years later, "while the sublicense was still in force, petitioners brought [a second] action" against National Screen, "alleg[ing] that the settlement of the [prior] suit was merely a device used by the

defendants in that case to perpetuate their conspiracy and monopoly." *Id.* at 324-25. National Screen moved to dismiss based on res judicata. *Id.* The district court granted the motion and the Third Circuit affirmed. *Id.* at 325-26.

The Supreme Court reversed. *Id.* at 330. The Supreme Court explained:

> That both suits involved "essentially the same course of wrongful conduct" is not decisive. Such a course of conduct—for example, an abatable nuisance—may frequently give rise to more than a single cause of action. And so it is here. ***The conduct presently complained of was all subsequent to the [prior] judgment*** …. Under these circumstances … the [prior] judgment does not constitute a bar to the instant suit.

Id. at 327-28 (emphasis added). This binding precedent is universally and consistently applied. *See, e.g.*, *Allegheny Int'l, Inc. v. Allegheny Ludlum Steel Corp.*, 40 F.3d 1416, 1429 (3d Cir. 1994) (relying on *Lawlor* to hold that res judicata did not bar claims for "reimbursement of expenses [plaintiff] had not even incurred at the time that its 1985 suit was dismissed with prejudice"); *Cellar Door Prods., Inc. of Mich. v. Kay*, 897 F.2d 1375, 1378 (6th Cir. 1990) (relying on *Lawlor* to hold that "causes of action that arose subsequent to the 1983 dismissal are not barred by res judicata").

This Court has made identical findings in the context of patent infringement actions. In *Brain Life*, this Court concluded that "claim preclusion [does] bar all allegations of infringement relating to activity that predated the [prior] judgment," but that claim preclusion cannot bar "infringement allegations [that] are temporally

limited to acts occurring after final judgment was entered in the first suit." 746 F.3d at 1053-54. This Court explained:

> [T]o the extent Brain Life's allegations of infringement are directed to products created and, most importantly, acts of alleged infringement occurring after entry of the final judgment in the [prior] [l]itigation, those claims are not barred by the doctrine of claim preclusion. Quite simply, Brain Life could not have asserted infringement claims against the products in question for acts of alleged infringement that postdate the final judgment in the [prior] [l]itigation.

Id. at 1054. This Court reached the same conclusion in *Aspex Eyewear, Inc. v. Marchon Eyewear, Inc.*, finding that infringement claims based on acts of infringement that occurred after the prior judgment were not barred because "res judicata requires that in order for a particular claim to be barred, it is necessary that the claim either was asserted, or could have been asserted, in the prior action." 672 F.3d 1335, 1342 (Fed. Cir. 2012). Relying on *Lawlor*, this Court explained:

> Under well-settled principles, a party who sues a tortfeasor is ordinarily not barred by a prior judgment from seeking relief for discrete tortious action by the same tortfeasor that occurs subsequent to the original action. That rule is based on the principle that res judicata requires a party to assert all claims that the party could have asserted in the earlier lawsuit; it follows that if the party could not have asserted particular claims—because the tortious conduct in question had not occurred at that time—those claims could not have been asserted and therefore are not barred by res judicata.

Id. (citing *Lawlor*, 349 U.S. at 328).

The same is true in this case. Mentor's claims that Synopsys infringes the '176 and '531 Patents are based exclusively on acts of infringement that occurred

after the Mentor-EVE License expired on October 4, 2012. A1216-29. These claims did not exist at the time of the 2006 Action and thus could not have been brought in the 2006 Action. Pursuant to *Lawlor*, *Brain Life*, and *Aspex*, these claims cannot be barred by res judicata and the district court's summary judgment order should be reversed.

CONCLUSION

The Court should grant Mentor's cross-appeal on issues of willfulness, res judicata, and invalidity of the '882 Patent and affirm the district court's judgment in all other respects.

Dated: September 30, 2015

Respectfully submitted,

/s/ *Mark E. Miller*

Mark E. Miller
O'Melveny & Myers LLP
Two Embarcadero Center, 28th Floor
San Francisco, California 94111
Telephone: (415) 984-8700
Facsimile: (415) 984-8701
markmiller@omm.com

Counsel for Plaintiff-Cross-Appellant
Mentor Graphics Corp.

ADDENDUM TO BRIEF OF APPELLEE AND CROSS-APPELLANT MENTOR GRAPHICS CORPORATION

Table of Contents

U.S. Patent 5,649,176

A400 - 426

US005649176A

United States Patent [19]

Selvidge et al.

[11] Patent Number: 5,649,176

[45] Date of Patent: Jul. 15, 1997

[54] **TRANSITION ANALYSIS AND CIRCUIT RESYNTHESIS METHOD AND DEVICE FOR DIGITAL CIRCUIT MODELING**

[75] Inventors: **Charles W. Selvidge**, Charlestown; **Matthew L. Dahl**, Cambridge, both of Mass.

[73] Assignee: **Virtual Machine Works, Inc.**, Cambridge, Mass.

[21] Appl. No.: **513,605**

[22] Filed: **Aug. 10, 1995**

[51] **Int. Cl.**[6] **G06F 1/12**

[52] **U.S. Cl.** **395/551**; 364/489

[58] **Field of Search** 395/551, 500; 364/488, 489, 490, 491

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,450,560	5/1984	Conner	371/25
4,697,241	9/1987	Lavi	364/488
5,180,937	1/1993	Laird et al.	327/276
5,420,544	5/1995	Ishibashi	331/11
5,428,626	6/1995	Frisch et al.	364/488 X

FOREIGN PATENT DOCUMENTS

0 453 171 A2	10/1991	European Pat. Off.	G06F 1/04
2 180 382	3/1987	United Kingdom	H03K 19/00

OTHER PUBLICATIONS

Laird, D., et al., "Delay Compensator," *LSI Logic Corp.*, pp. 1–8, (Aug. 1990).

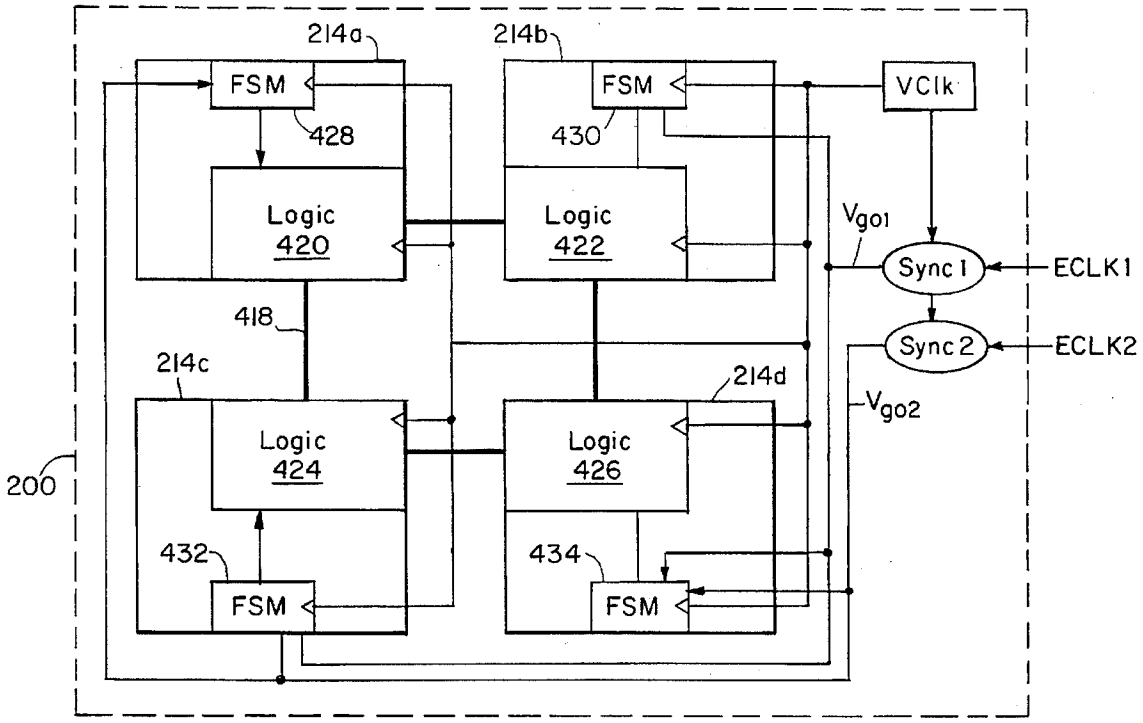
Primary Examiner—Thomas M. Heckler

Attorney, Agent, or Firm—Hamilton, Brook, Smith & Reynolds, P.C.

[57] **ABSTRACT**

A method of configuring a configurable logic system, including a single or multi-FPGA network, is disclosed in which an internal clock signal is defined that has a higher frequency than timing signals the system receives from the environment in which it is operating. The frequency can be at least ten times higher than a frequency of the environmental timing signals. The logic system is configured to have a controller that coordinates operation of its logic operation in response to the internal clock signal and environmental timing signals. Specifically, the controller is a finite state machine that provides control signals to sequential logic elements such as flip-flops. The logic elements are clocked by the internal clock signal. In the past, emulation or simulation devices, for example, operated in response to timing signals from the environment. A new internal clock signal, invisible to the environment, rather than the timing signals is used to control the internal operations of the devices. Additionally, a specific set of transformations are disclosed that enable the conversion of a digital circuit design with an arbitrary clocking methodology into a single clock synchronous circuit.

50 Claims, 14 Drawing Sheets



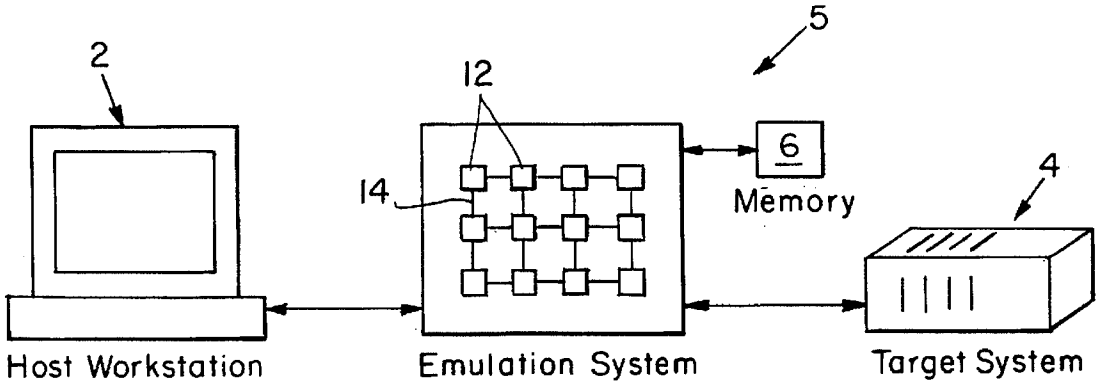


FIG. 1
(Prior Art)

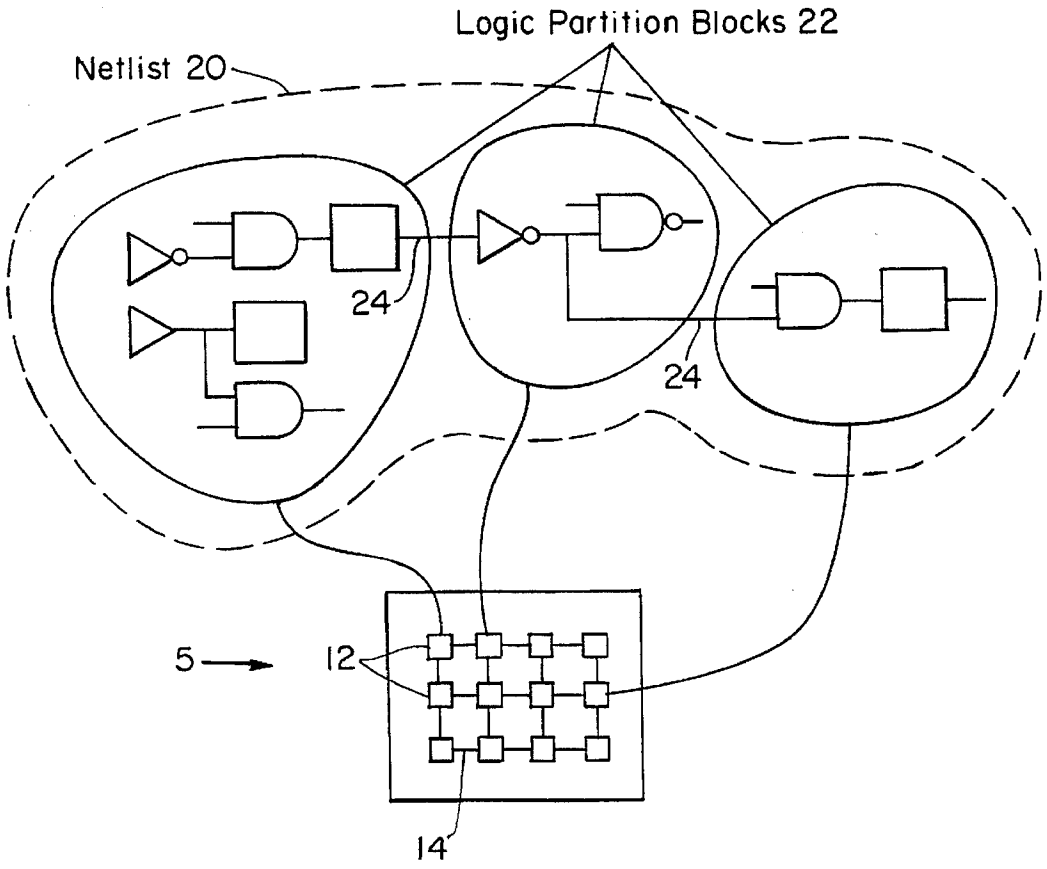


FIG. 2

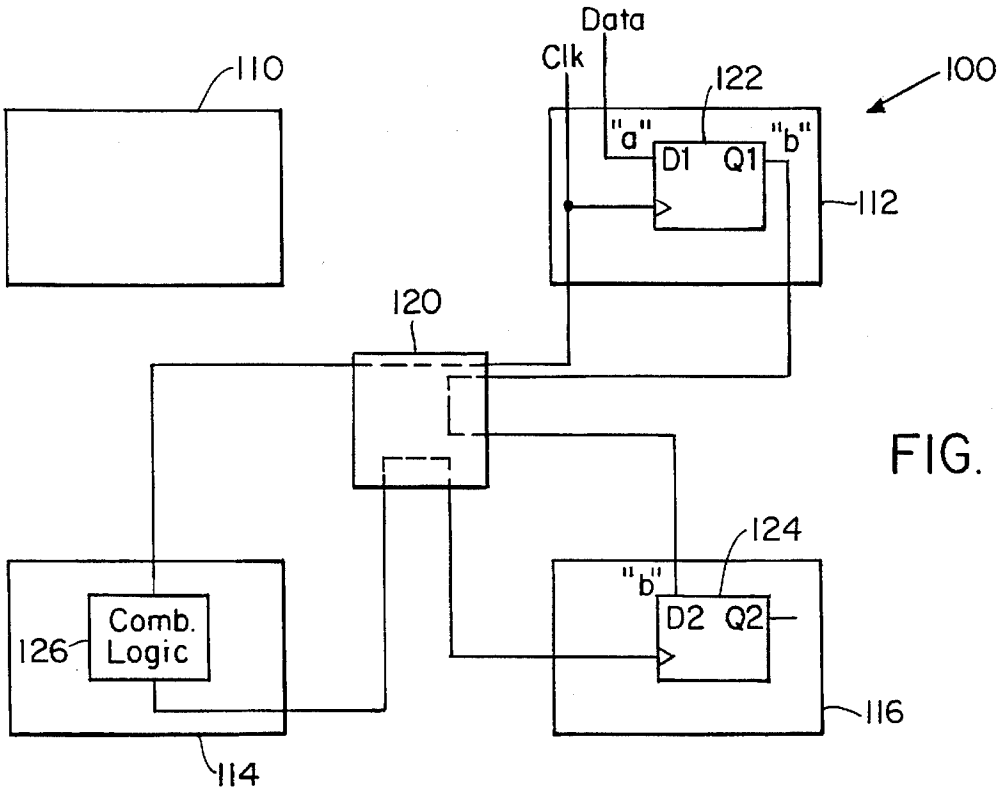

Data
Clk
110
122
100
"a"
"b"
D1
Q1
112
120
124
"b"
D2
Q2
Comb.
Logic
126
116
114

FIG. 3

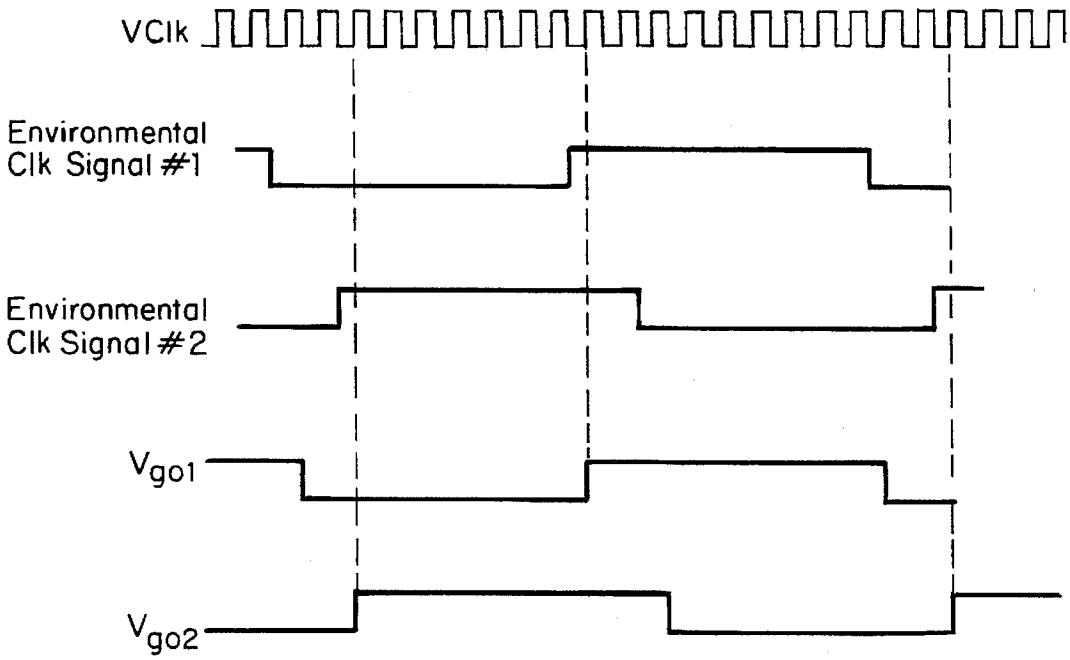

VClk
Environmental
Clk Signal #1
Environmental
Clk Signal #2
V_{go1}
V_{go2}

FIG. 4B

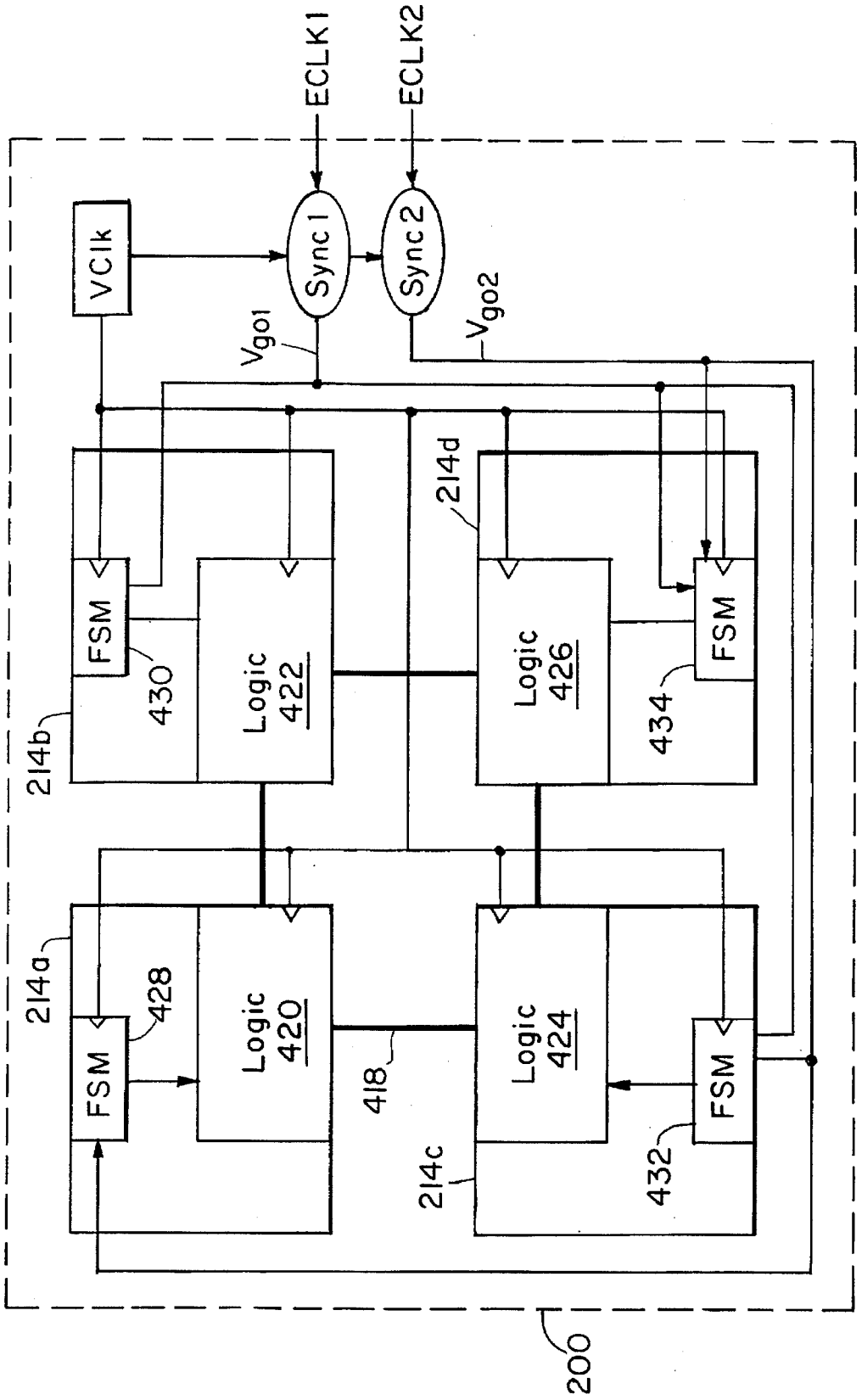

ECLK1
ECLK2
VClk
Sync 1
Sync 2
Vgo1
Vgo2
214d
FSM
430
Logic 422
214b
Logic 426
FSM
434
214a
428
FSM
Logic 420
418
Logic 424
FSM
214c
432
200

FIG. 4A

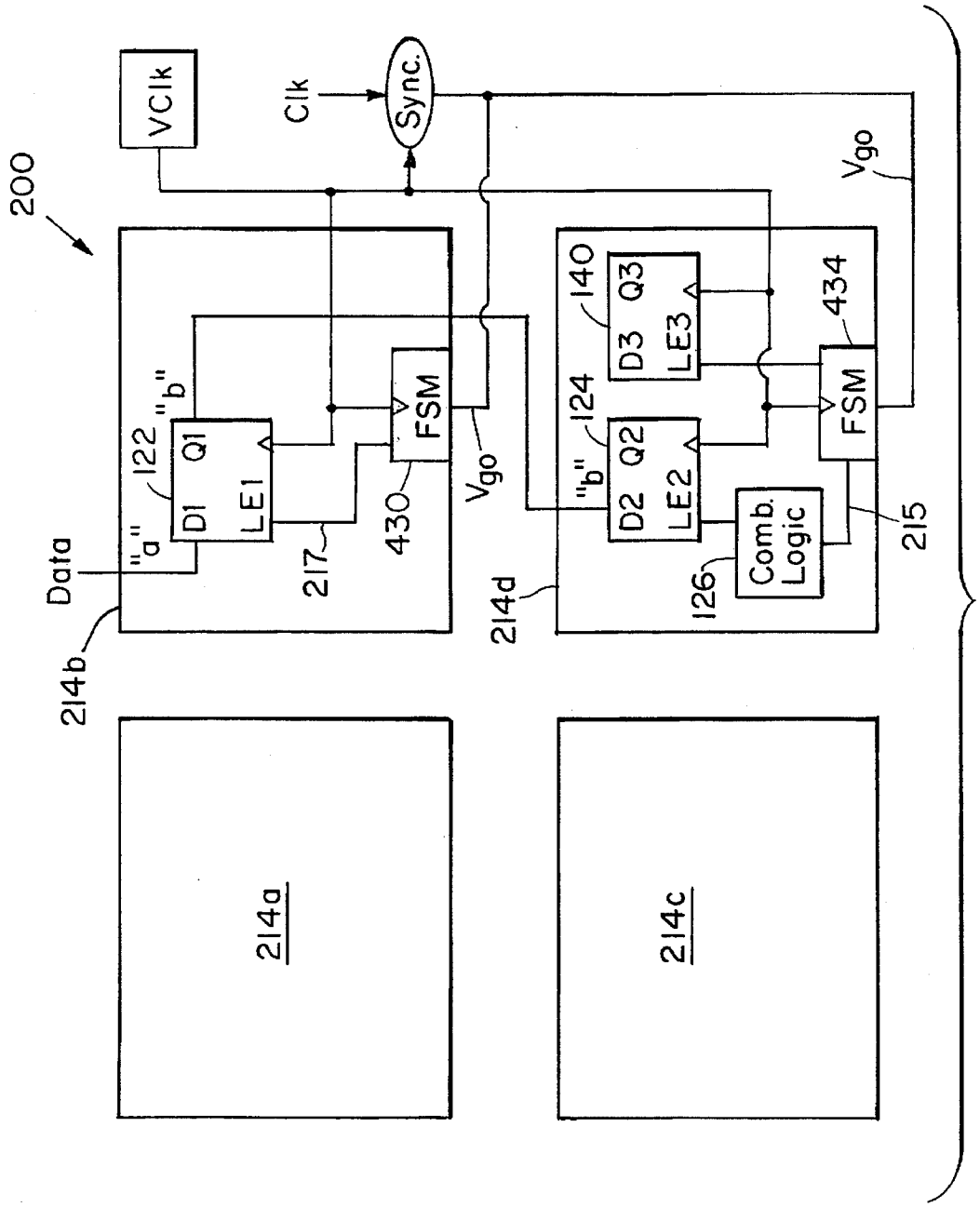

200
VClk
Clk
Sync.
Vgo
140
D3 Q3
LE3
434
FSM
124
"b"
D2 Q2
LE2
Comb. Logic
215
126
214d
"b"
122
D1 Q1
LE1
FSM
430
217
Vgo
Data
"a"
214b
214a
214c

FIG. 5A

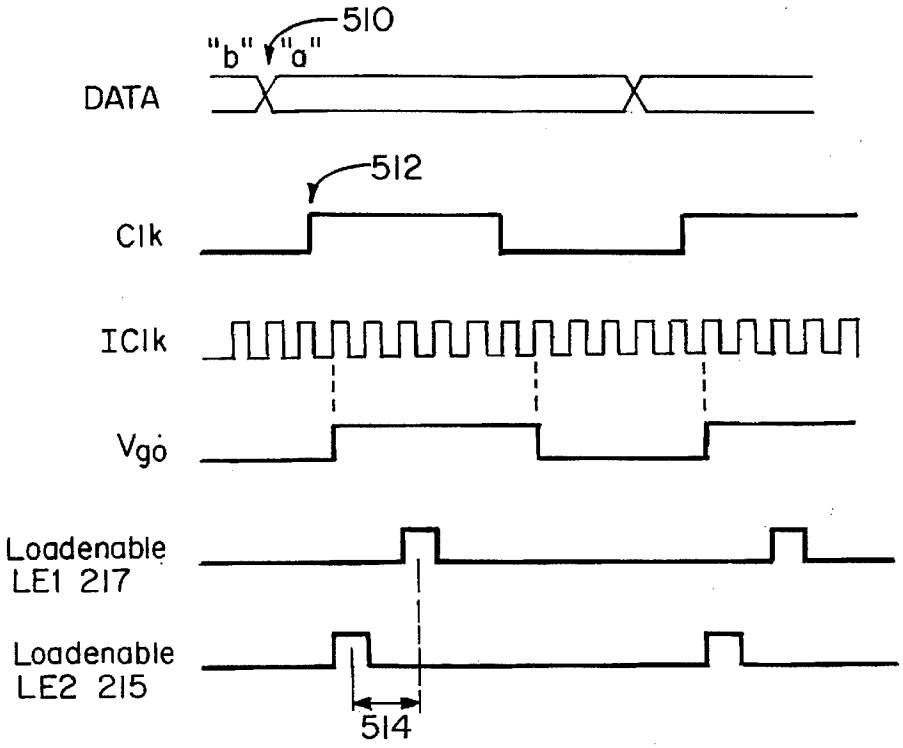

510
"b"
"a"
DATA
512
Clk
IClk
Vgo
Loadenable
LE1 217
Loadenable
LE2 215
514

FIG. 5B

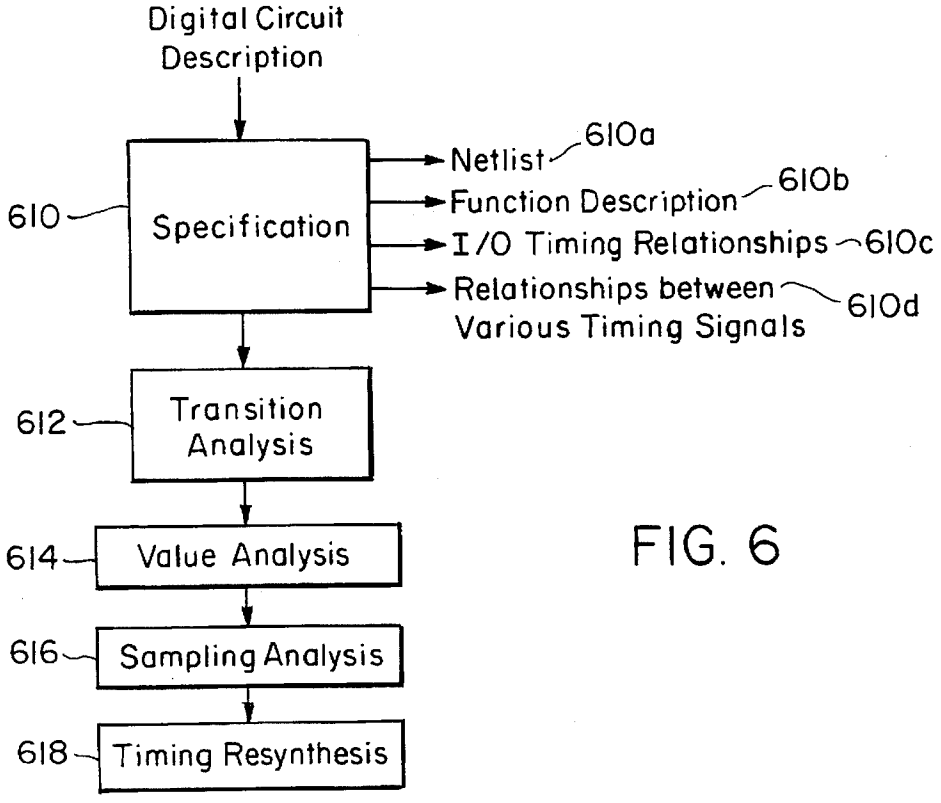

Digital Circuit
Description
610
Specification
Netlist
610a
Function Description
610b
I/O Timing Relationships
610c
Relationships between
Various Timing Signals
610d
612
Transition
Analysis
614
Value Analysis
616
Sampling Analysis
618
Timing Resynthesis

FIG. 6

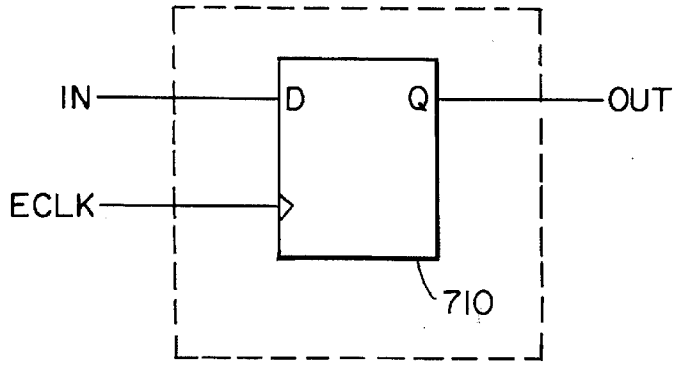

IN
D
Q
OUT
ECLK
710

FIG. 7A

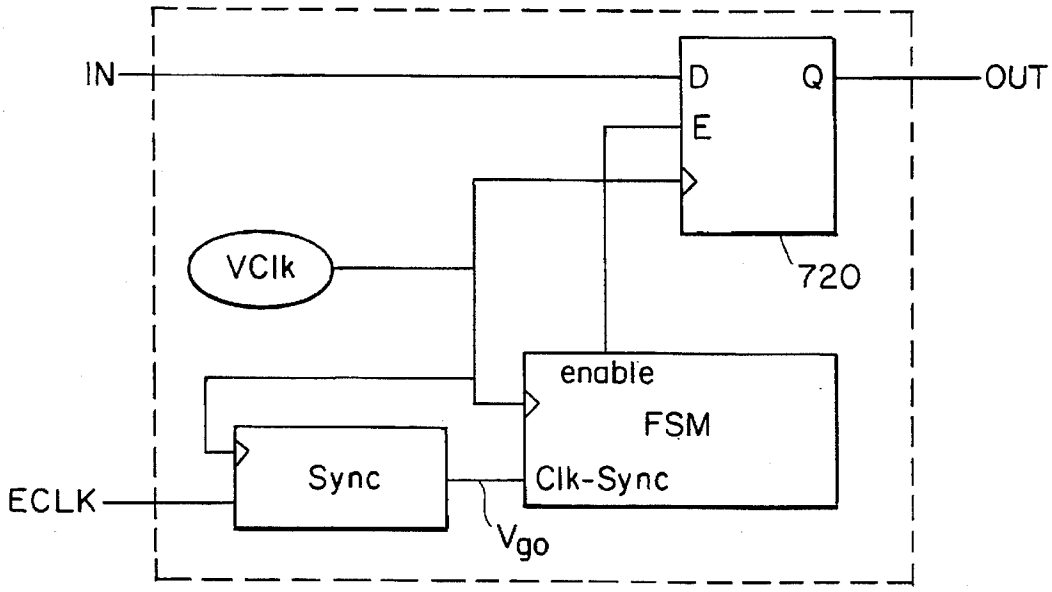

IN
D
Q
OUT
E
VClk
720
enable
FSM
Sync
Clk-Sync
ECLK
Vgo

FIG. 7B

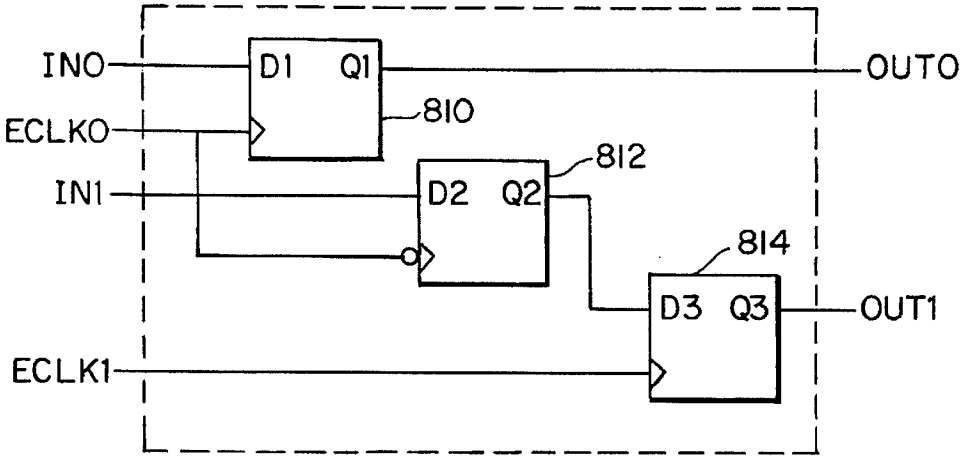

INO
ECLKO
IN1
ECLK1
D1 Q1
810
D2 Q2
812
D3 Q3
814
OUTO
OUT1

FIG. 8A

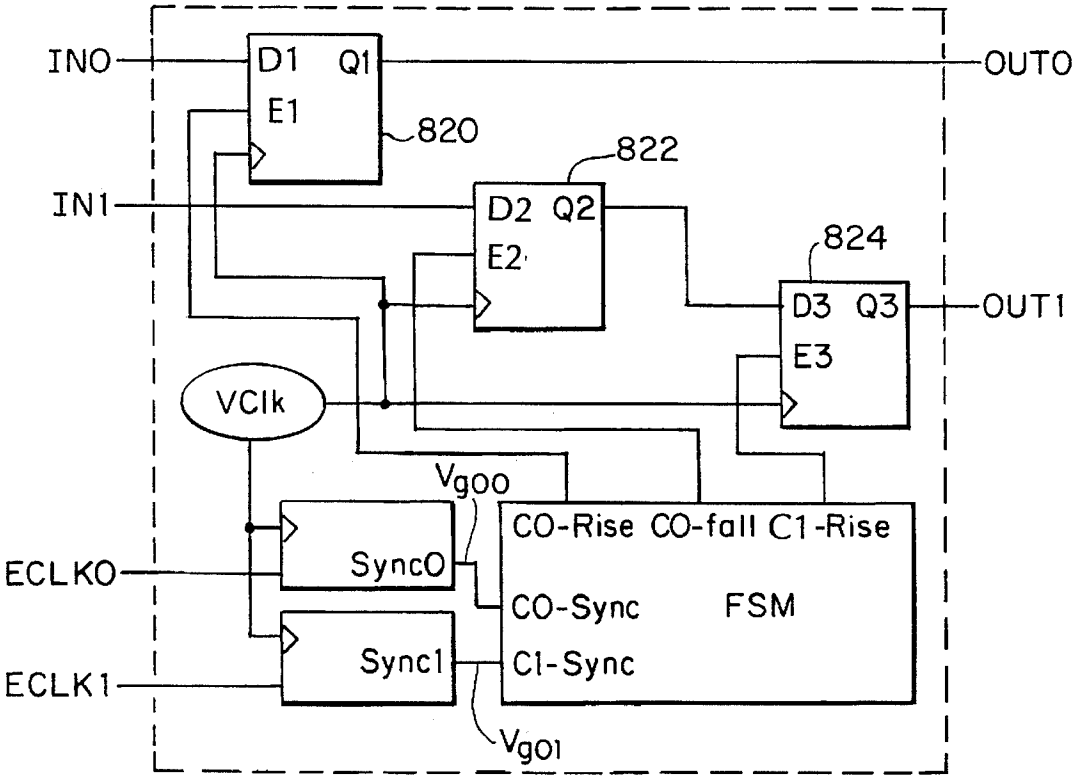

INO
IN1
ECLKO
ECLK1
D1 Q1
E1
820
D2 Q2
E2
822
D3 Q3
E3
824
OUTO
OUT1
VClk
Vgoo
Vgo1
SyncO
Sync1
CO-Rise CO-fall C1-Rise
CO-Sync
C1-Sync
FSM

FIG. 8B

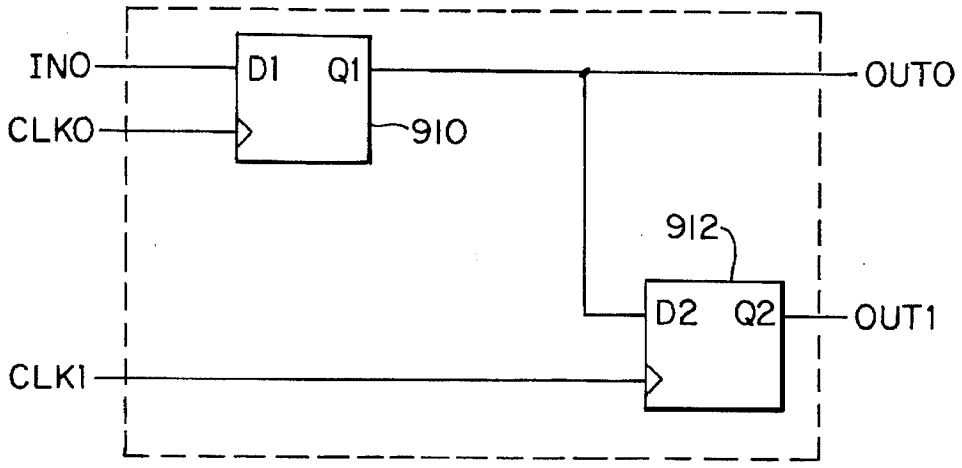

INO
CLKO
D1
Q1
910
OUTO
912
D2
Q2
OUT1
CLK1

FIG. 9A

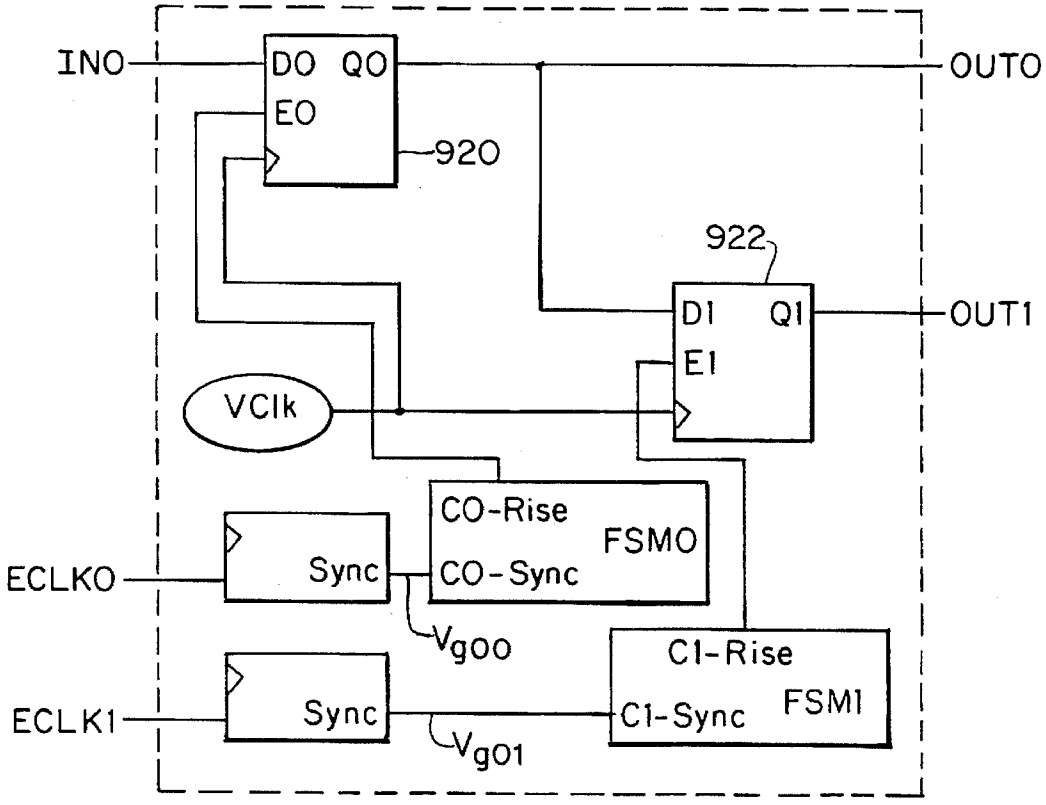

INO
DO
QO
EO
920
OUTO
922
D1
Q1
E1
OUT1
VClk
CO-Rise
FSMO
ECLKO
Sync
CO-Sync
Vgoo
C1-Rise
ECLK1
Sync
C1-Sync
FSM1
Vgo1

FIG. 9B

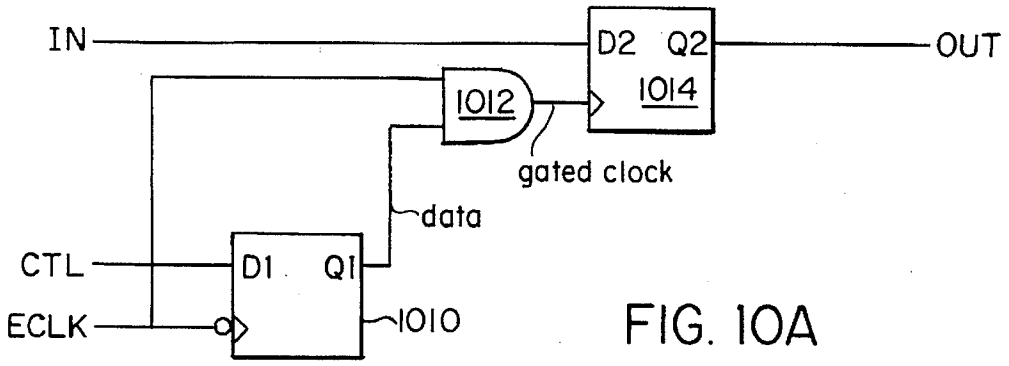


FIG. 10A

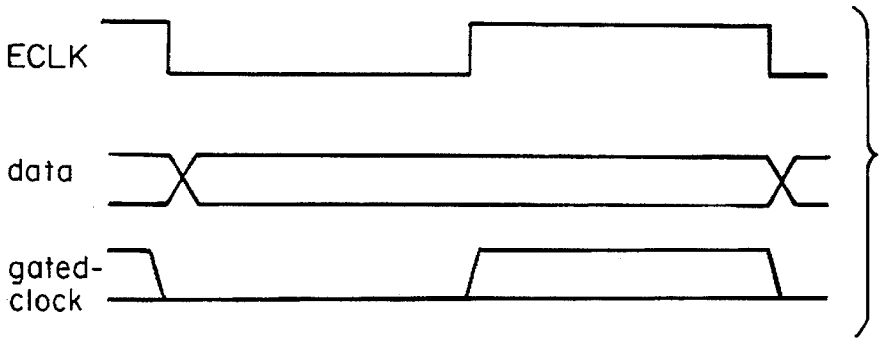


FIG. 10B

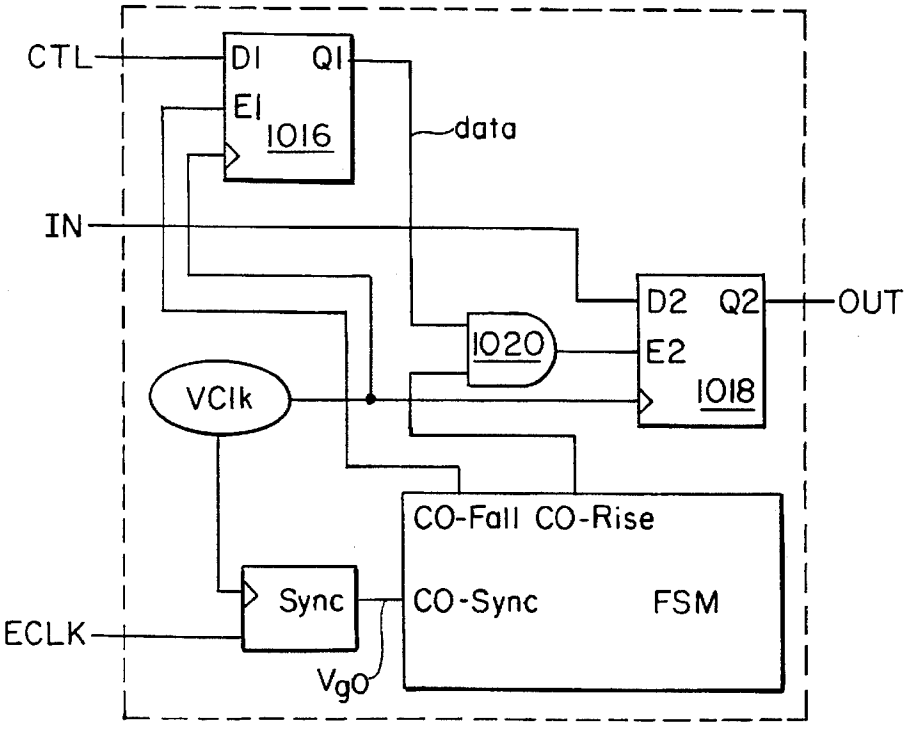


FIG. 10C

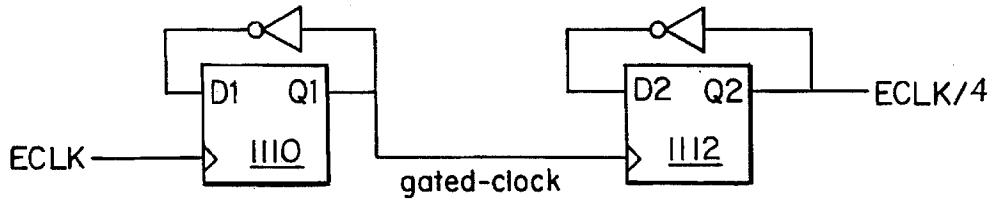


FIG. 11A

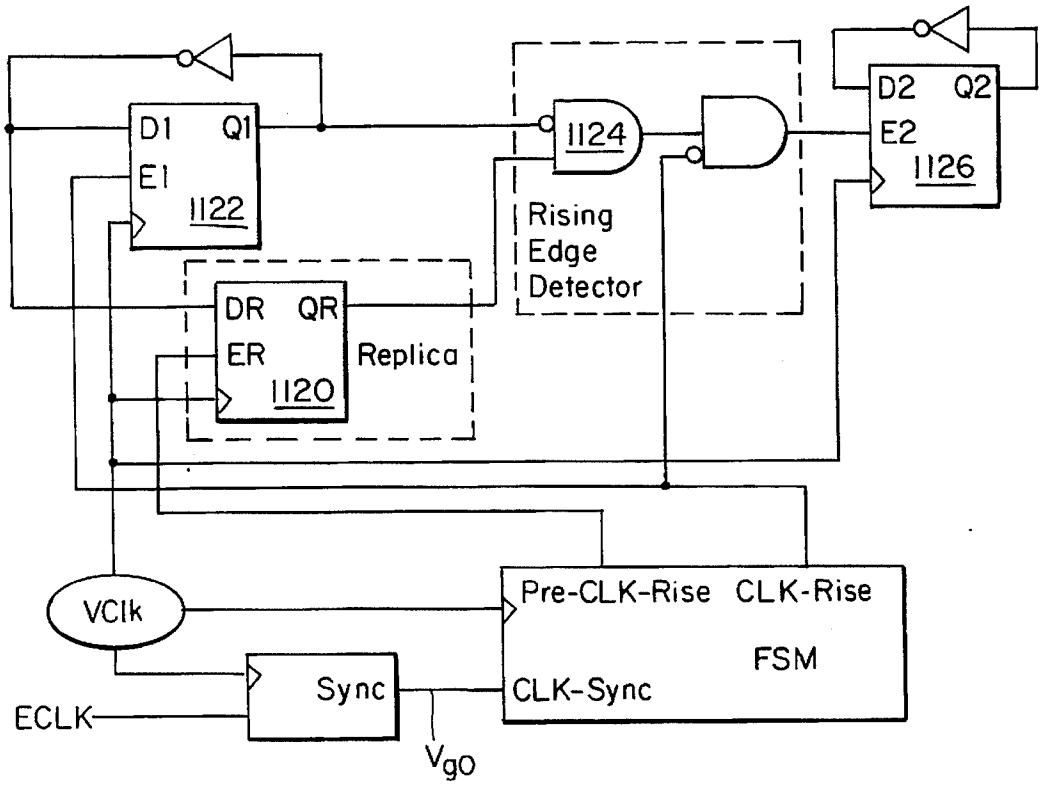


FIG. 11B

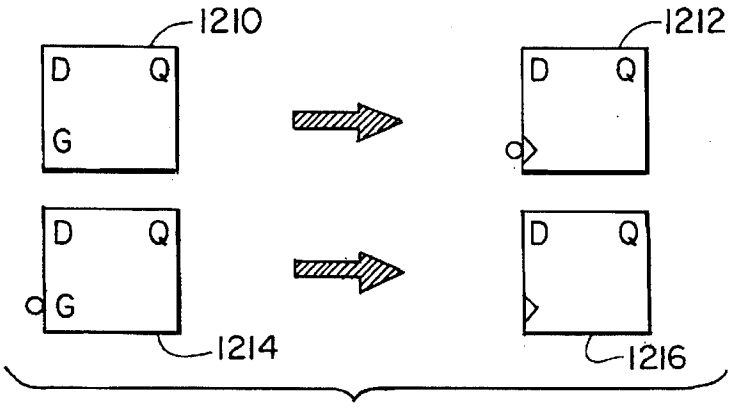


FIG. 12

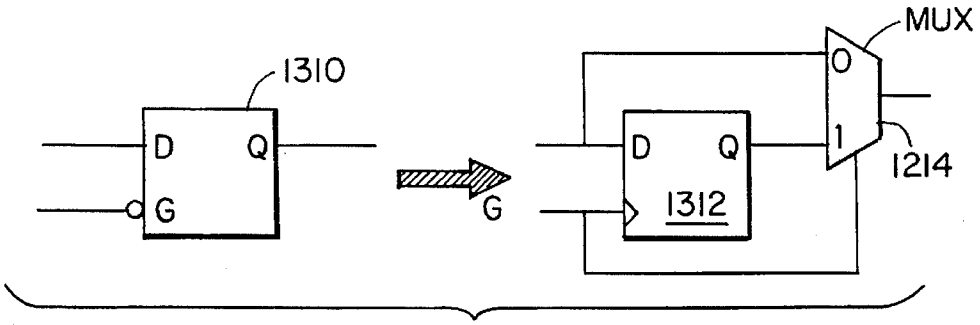


FIG. 13

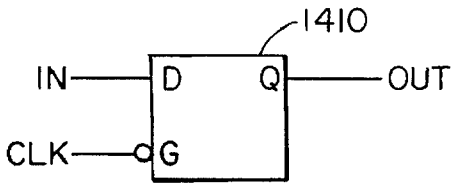


FIG. 14A

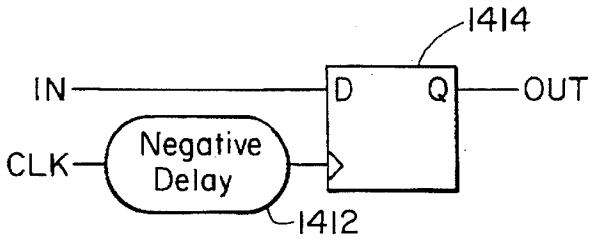


FIG. 14B

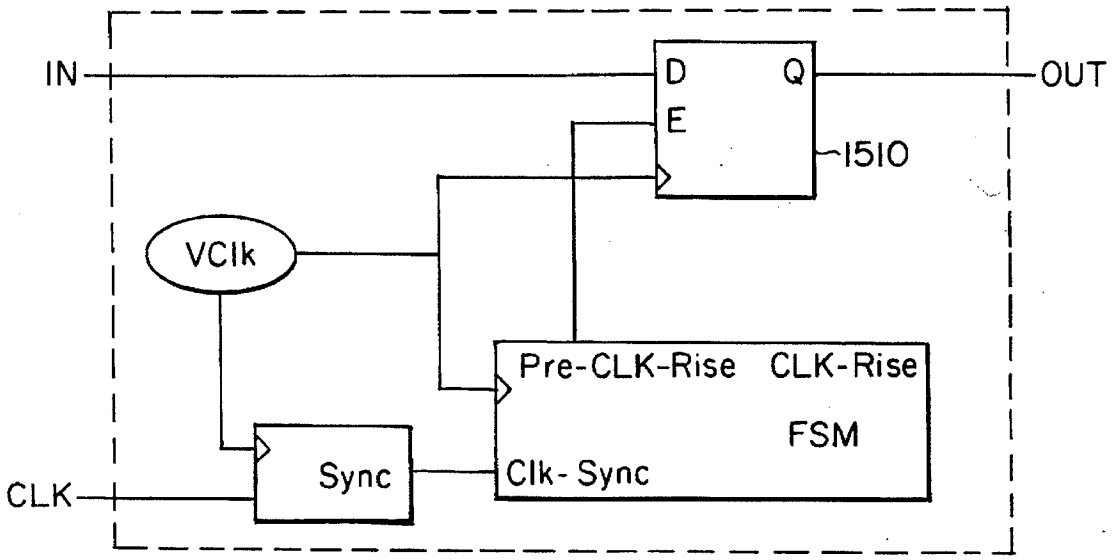


FIG. 15

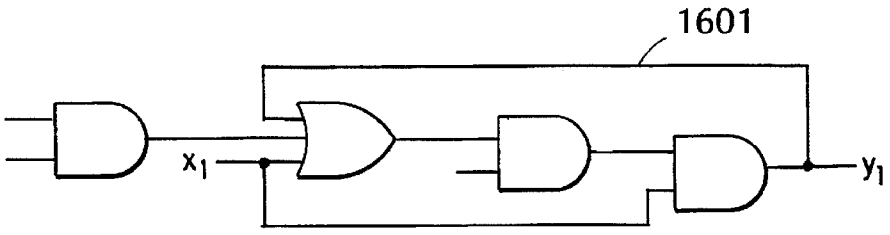


FIG. 16A

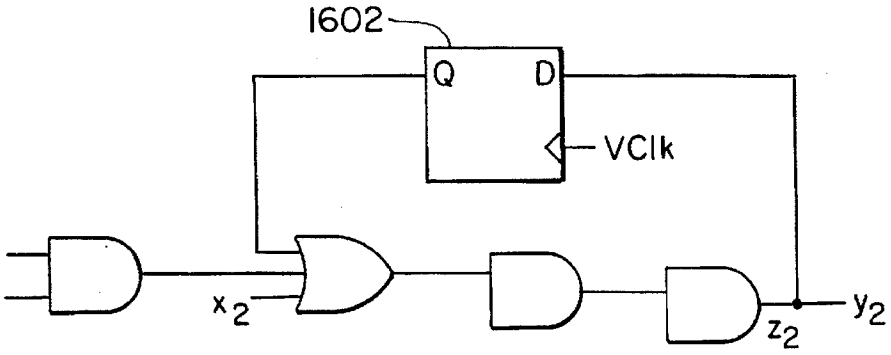


FIG. 16B

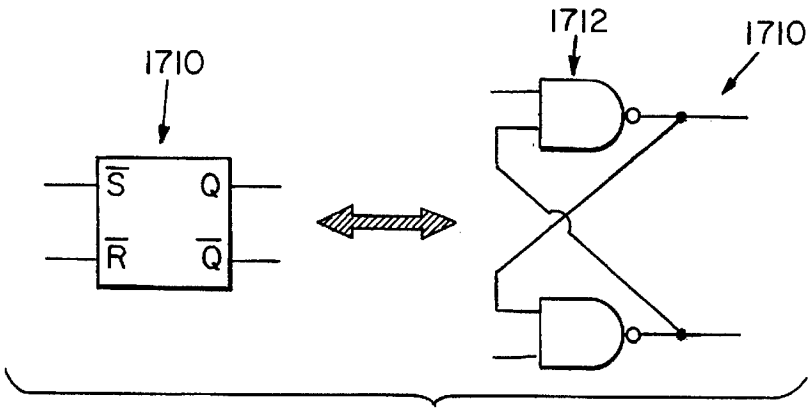


FIG. 17A

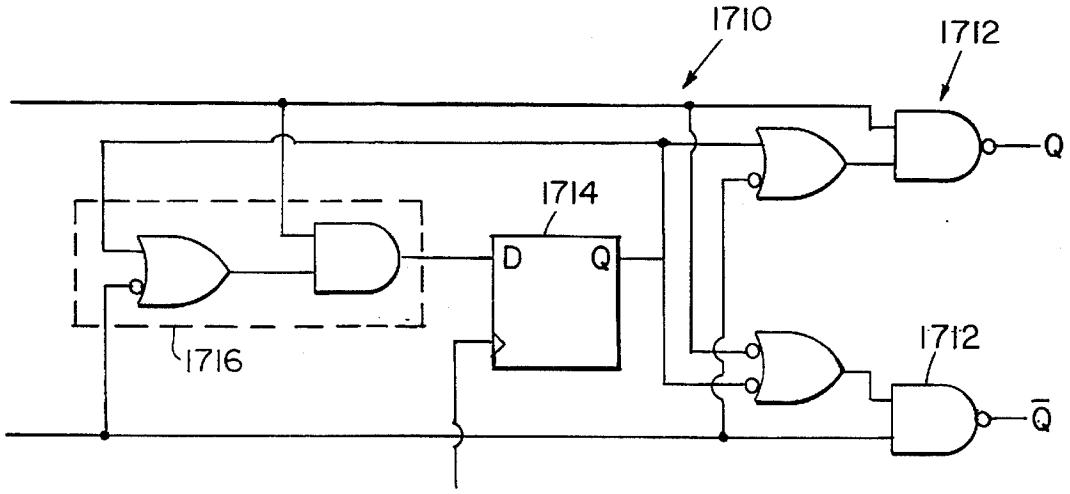


FIG. 17B

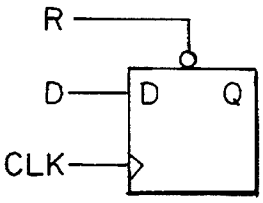


FIG. 18A

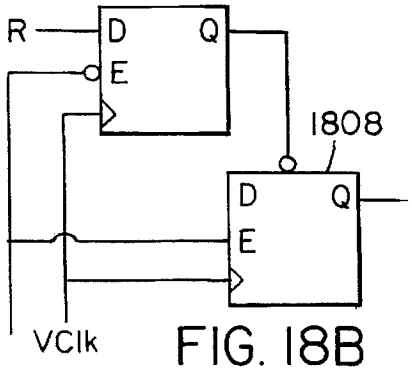


FIG. 18B

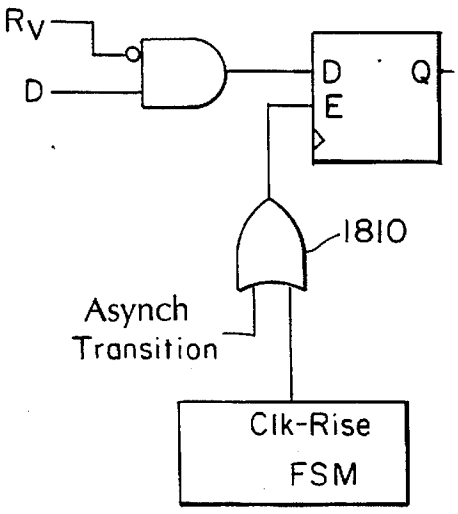


FIG. 18C

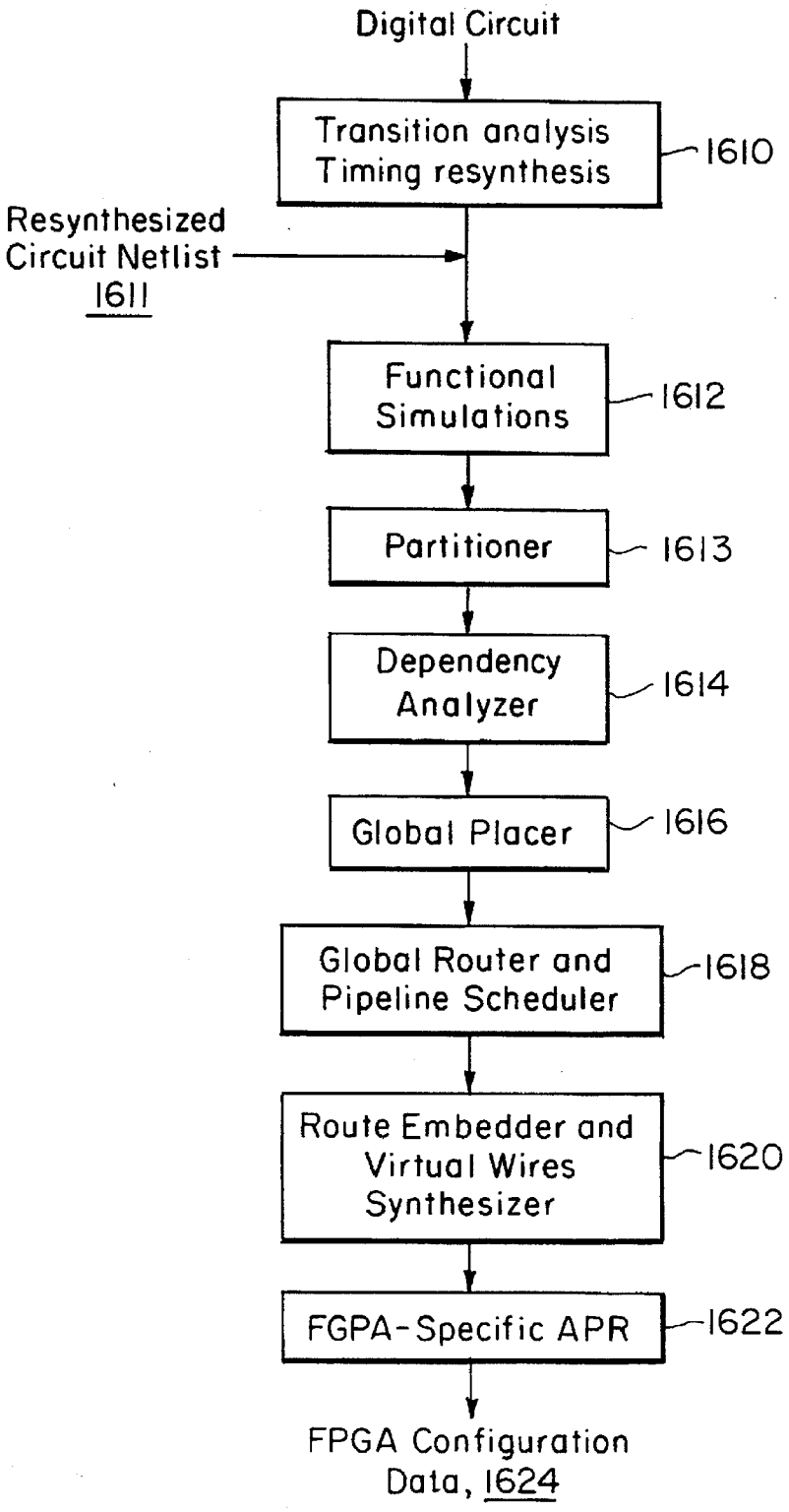

Digital Circuit
Transition analysis
Timing resynthesis
1610
Resynthesized
Circuit Netlist
1611
Functional
Simulations
1612
Partitioner
1613
Dependency
Analyzer
1614
Global Placer
1616
Global Router and
Pipeline Scheduler
1618
Route Embedder and
Virtual Wires
Synthesizer
1620
FGPA-Specific APR
1622
FPGA Configuration
Data, 1624

FIG. 19

TRANSITION ANALYSIS AND CIRCUIT RESYNTHESIS METHOD AND DEVICE FOR DIGITAL CIRCUIT MODELING

BACKGROUND OF THE INVENTION

Configurable logic devices are a general class of electronic devices that can be easily configured to perform a desired logic operation or calculation. One example is Mask Programmed Gate Arrays (MPGA). These devices offer density and performance. Poor turn around time coupled with only one-time configurability tend to diminish their ubiquitous use. Reconfigurable logic devices or programmable logic devices (such as Field Programmable Gate Arrays (FPGA)) offer lower levels of integration but are reconfigurable, i.e., the same device may be programmed many times to perform different logic operations. Most importantly, the devices can be programmed to create gate array prototypes instantaneously, allowing complete dynamic reconfigurability, something that MPGAs can not provide.

System designers commonly use reconfigurable logic devices such as FPGAs to test logic designs prior to manufacture or fabrication in an effort to expose design flaws. Usually, these tests take the form of emulations in which a reconfigurable logic devices models the logic design, such as a microprocessor, in order to confirm the proper operation of the logic design along with possibly its compatibility with an environment or system in which it is intended to operate.

In the case of testing a proposed microprocessor logic design, a netlist describing the internal architecture of the microprocessor is compiled and then loaded into a particular reconfigurable logic device by some type of configuring device such as a host workstation. If the reconfigurable logic device is a single or array of FPGAs, the loading step is as easy as down-loading a file describing the compiled netlist to the FPGAs using the host workstation or other computer. The programmed configurable logic device is then tested in the environment of a motherboard by confirming that its response to inputs agrees with the design criteria.

Alternatively, reconfigurable logic devices also find application as hardware accelerators for simulators. Rather than testing a logic design by programming a reconfigurable device to "behave" as the logic device in the intended environment for the logic design, e.g., the motherboard, a simulation involves modeling the logic design on a workstation. In this environment, the reconfigurable logic device performs gate evaluations for portions of the model in order to relieve the workstation of this task and thereby decreases the time required for the simulation.

Recently, most of the attention in complex logic design modeling has been directed toward FPGAs. The lower integration of the FPGAs has been overcome by forming heterogeneous networks of special purpose FPGA processors connected to exchange signals via some type of interconnect. The network of the FPGAs is heterogeneous not necessarily in the sense that it is composed of an array of different devices but that the devices have been individually configured to cooperatively execute different sections, or partitions, of the overall logic design. These networks rely on static routing at compile-time to organize the propagation of logic signals through the FPGA network. Static refers to the fact that all data or logic signal movement can be determined and optimized during compiling.

When a logic design intended for eventual MPGA fabrication is mapped to an FPGA, hold time errors are a problem that can arise, particularly in these complex configurable

logic device networks. A digital logic design that has been loaded into the configurable logic devices receives timing signals, such as clock signals, and data signals from the environment in which it operates. Typically, these timing signals coordinate the operation of storage or sequential logic components such as flip-flops or latches. These storage devices control the propagation of combinational signals, which are originally derived from the environmental data signals, through the logic devices.

Hold time problems commonly arise where a timing signal is intended to clock a particular storage element to signal that a value at the element's input terminal should be held or stored. As long as the timing signal arrives at the storage element while the value is valid, correct operation is preserved. Hold time violations occur when the timing signal is delayed beyond a time for which the value is valid, leading to the loss of the value. This effect results in the destruction of information and generally leads to the improper operation of the logic design.

Identification and mitigation of hold time problems presents many challenges. First, while the presence of a hold time problem can be recognized by the improper operation of the logic design, identifying the specific location within the logic design of the hold time problem is a challenge. This requires sophisticated approximations of the propagation delays of timing signals and combinational signals through the logic design. Once a likely location of a hold time problem has been identified, the typical approach is somewhat ad hoc. Delay is added on the path of the combinational signals to match the timing signal delays. This added delay, however, comes at its own cost. First, the operational speed of the design must now take into account this new delay. Also, new hold time problems can now arise because of the changed clock speed. In short, hold time problems are both difficult to identify and then difficult to rectify.

Other problems arise when a logic design intended for ultimate MPGA fabrication, for example, is realized in FPGAs. Latches, for instance, are often implemented in MPGAs. FPGA, however, do not offer a corresponding element.

SUMMARY OF THE INVENTION

The present invention seeks to overcome the hold time problem by imposing a new timing discipline on a given digital circuit design through a resynthesis process that yields a new but equivalent circuit. The resynthesis process also transforms logic devices and timing structures to those that are better suited to FPGA implementation. This new timing discipline is insensitive to unpredictable delays in the logic devices and eliminates hold time problems. It also allows efficient implementation of latches, multiple clocks, and gated clocks. By means of the resynthesis, the equivalent circuit relies on a new higher frequency internal clock (or virtual clock) that is distributed with minimal skew. The internal clock signal controls the clocking of all or substantially all the storage elements, e.g. flip-flops, in the equivalent circuit, in effect discretizing time and space into manageable pieces. The user's clocks are treated in the same manner as user data signals.

In contrast with conventional approaches, the present invention does not allow continuous inter-FPGA signal flow. Instead, all signal flow is synchronized to the internal clock so that signals flow between flip-flops through intermediate FPGAs in discrete hops. The internal clock provides a time base for the circuit's operation.

In general, according to one aspect, the invention features a method of configuring a configurable or programmable

logic system. Generally, such logic systems include single or multi-FPGA network, although the invention can be applied to other types of configurable devices. Particular to the invention, the logic system is provided with an internal clock signal that typically has a higher frequency, by a factor of at least four, than timing signals the system receives from the environment in which it is operating. The logic system is configured to have a controller that coordinates operation of the logic in response to the internal clock signal and the environmental timing signals. In the past, while emulation or simulation devices, for example, operated in response to timing signals from the environment, a new internal clock signal, invisible to the environment, was not used to control the internal operations of the devices.

In specific embodiments, a synchronizer is incorporated to essentially generate a synchronized version of the environmental timing signal. This synchronized version behaves much like other data signals from the environment. This synchronizer feeds the resulting sampled environmental clock signals to a finite state machine, which generates control signals. The logic operations are then coordinated by application of these control signals to sequential logic elements.

In more detail, the logic system is configured to have both combinational logic, e.g. logic gates, and sequential logic, e.g. flip-flops, to perform the logic operations. The control signals function as load enable signals to the sequential logic. The internal clock signal is received at the clock terminals of that logic. Just like the original digital circuit design, each sequential logic element operates in response to the environmental timing signals. Now, however, these timing signal control the load enable of the elements, not the clocking. It is the internal clock signal that now clocks the elements. As a result, the resynthesized circuit operates synchronously with a single clock signal regardless of the clocking scheme of the original digital circuit.

In general, according to another aspect, the invention features a method for converting a digital circuit design into a new circuit that is substantially functionally equivalent to the digital circuit design. First, the internal clock signal is defined, then sequential logic elements of the digital circuit design are resynthesized to operate in response to the internal clock signal in the new circuit rather than simply the environmental timing signals.

In specific embodiments, flip-flops of the digital circuit design, which are clocked by the environmental timing signal, are resynthesized to be clocked by the internal clock signal and load enabled in response to the environmental timing signals. Finite state machines are used to actually generate control signals that load enable each flip-flop. The load enable signals are sometimes also generated from a logic combination of finite state machine signals and logic gates.

In other embodiments, latches in the digital circuit design, which were gated by the environmental timing signals, are resynthesized to be flip-flops or latches in future FPGA designs in the new circuit that are clocked by the new internal or virtual clock signal. These new flip-flops are load enabled in response to the environmental timing signals.

In general, according to still another aspect, the invention features a logic system for generating output signals to an environment in response to at least one environmental timing signal and environmental data signals provided from the environment. This logic system has its own internal clock and at least one configurable logic device. The internal architecture of the configurable device includes logic for generating the output signals in response to the environmental data signals and a controller, specifically a finite state machine, for coordinating operation of the logic in response to the internal clock signal and the environmental timing signal.

Specifically, the logic includes sequential and combinational logic elements. The sequential logic elements are clocked by the internal clock signal and load enabled in response to the environmental timing signals.

The above and other features of the invention including various novel details of construction and combinations of parts, and other advantages, will now be more particularly described with reference to the accompanying drawings and pointed out in the claims. It will be understood that the particular method and device embodying the invention is shown by way of illustration and not as a limitation of the invention. The principles and features of this invention may be employed in various and numerous embodiments without the departing from the scope of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

In the accompanying drawings, like reference characters refer to the same parts throughout the different views. The drawings are not necessarily to scale and in some cases have been simplified. Emphasis is instead placed upon illustrating the principles of the invention. Of the drawings.

FIG. **1** is a schematic diagram showing a prior art emulation system and its interaction with an environment and a host workstation;

FIG. **2** shows a method for impressing a logic design on the emulation system;

FIG. **3** is a schematic diagram of a configurable logic system that comprises four configurable logic devices—a portion of the internal logic structure of these devices has been shown to illustrate the origins of hold time violations;

FIG. **4A** is a schematic diagram of the logic system of the present invention showing the internal organization of the configurable logic devices and the global control of the logic devices by the internal or virtual clock;

FIG. **4B** is a timing diagram showing the timing relationships between the internal or virtual clock signal, environmental timing signals, and control signals generated by the logic system;

FIG. **5A** is a schematic diagram of a logic system of the present invention that comprises four configurable logic devices, the internal structure of these devices is the functional equivalent of the structure shown in FIG. **3** except that the circuit has been resynthesized according to the principles of the present invention;

FIG. **5B** is a diagram showing the timing relationship between the signals generated in the device of FIG. **5A**;

FIG. **6** illustrates a method by which a digital circuit description having an arbitrary clocking methodology is resynthesized into a functionally equivalent circuit that is synchronous with a single internal clock;

FIGS. **7A** and **7B** illustrate a timing resynthesis circuit transformation in which an edge-triggered flip-flop is converted into a load-enable type flip-flop;

FIGS. **8A** and **8B** illustrate a timing resynthesis circuit transformation in which a plurality of edge triggered flip-flops clocked by two phase-locked clock signals are converted into load enable flip-flops that are synchronous with the internal clock signal;

FIGS. **9A** and **9B** illustrate a timing resynthesis circuit transformation in which two edge triggered flip-flops

clocked by two arbitrary clock signals are transformed into load enabled flip-flops that operate synchronously with the internal clock signal;

FIGS. **10**A, **10**B, and **10**C illustrate a timing resynthesis circuit transformation in which two edge-triggered flip-flops, one of which is clocked by a gated clock, are transformed into two load-enable flip-flops that operate synchronously with the internal clock signal, FIG. **10**B is a timing diagram showing the signal values over time in the circuit;

FIG. **11**A and **11**B illustrate a timing resynthesis circuit transformation in which a complex gated clock structure, with a second flip-flop being clocked by a gated clock, is converted into a circuit containing three flip-flops and an edge detector, all of the flip-flops operating off of the internal clock signal in the new circuit;

FIG. **12** illustrates circuit transformations in which gated latches are converted into edge-triggered flip-flops on the assumption that the latches are never sampled when open, i.e., latch output is not registered into another storage element when they are open;

FIG. **13** illustrates a timing resynthesis circuit transformation in which a gated latch is converted into an edge-triggered flip-flop and a multiplexor;

FIGS. **14**A and **14**B illustrate a timing resynthesis circuit transformation in which a latch is converted to an edge-triggered flip-flop with a negative delay at the clock input terminal to avoid glitches;

FIG. **15** illustrates a timing resynthesis circuit transformation of the negative delay flip-flop of FIG. **14**B into a flip-flop that operates synchronously with the internal clock signal;

FIGS. **16**A and **16**B illustrate a timing resythesis circuit transformation in which a flip-flop is inserted in a combinational loop to render the circuit synchronous with the virtual clock;

FIGS. **17**A and **17**B illustrate a timing resynthesis circuit transformation in which an RS flip-flop is transformed into a device that is synchronous with the virtual clock;

FIG. **18**A, **18**B, and **18**C illustrate a timing resynthesis circuit transformation for handling asynchronous preset and clears of state elements; and

FIG. **19** illustrates the steps performed by a compiler that resynthesizes the digital circuit design and converts it into FPGA configuration data that is loaded into the logic system **200**.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Turning now to the drawings, FIG. **1** is a schematic diagram showing an emulation system **5** of the prior art. The emulation system **5** operates in an environment such as a target system **4** from which it receives environmental timing signals and environmental data signals and responsive to these signals generates output data signals to the environment. A configuring device **2** such as a host workstation is provided to load configuration data into the emulation system **5**.

The emulation system **5** is usually constructed from individual configurable logic devices **12**, specifically FPGA chips are common. The configurable logic devices **12** are connected to each other via an interconnect **14**. Memory elements **6** are also optionally provided and are accessible by the configurable logic devices **12** through the interconnect **14**.

The host workstation **2** downloads the configuration data that will dictate the internal configuration of the logic devices **12**. The configuration data is compiled from a digital circuit description that includes the desired manner in which the emulation system **5** is intended to interact with the environment or target system **4**. Typically, the target system **4** is a larger electronic system for which some component such as a microprocessor is being designed. The description applies to this microprocessor and the emulation system **5** loaded with the configuration data confirms compatibility between the microprocessor design and the target system **4**. Alternatively, the target system **4** can be a device for which the logic system satisfies some processing requirements. Further, the emulation system **5** can be used for simulations in a software or FPGA based logic simulation.

FIG. **2** illustrates how the logic design is distributed among the logic devices **12** of the logic system **5**. A netlist **20** describing the logic connectivity of the logic design is separated into logic partition blocks **22**. The complexity of the blocks **22** is manipulated so that each can be realized in a single FPGA chip **12**. The logic signal connections that must bridge the partition blocks **24**, global links, are provided by the interconnect **14**. Obviously, the exemplary netlist **20** has been substantially simplified for the purposes of this illustration.

FIG. **3** illustrates the origins of hold time problems in conventional logic designs. The description is presented in the specific context of a configurable system **100**, such as an emulation system, comprising four configurable logic devices **110**–**116**, such as FPGAs, which are interconnected via a crossbar **120** interconnect. A portion of the internal logic of these devices is shown to illustrate the distribution of a gated clock and the potential problems from the delay of the clock.

The second logic device **112** has been programmed with a partition of the intended logic design that includes an edge-triggered D-type flip-flop **122**. This flip-flop **122** receives a data signal DATA at an input terminal D1 and is clocked by a clock signal CLK, both of which are from the environment in which the system **100** is intended to operate. The output terminal Q1 of the first flip-flop is connected to a second flip-flop **124** in the fourth logic device **116** through the crossbar **120**. This second flip-flop **124** is also clocked by the clock signal, albeit a gated version that reaches the second flip-flop **124** through the crossbar **120**, through combinational logic **126** on a third configurable logic device **114** and through the crossbar **120** a second time before it reaches the clock input of the second flip-flop **124**.

Ideally, the rising edge of the clock signal should arrive at both the first flip-flop **122** and the second flip-flop **124** at precisely the same time. As a result of this operation, the logic value "b" held at the output terminal Q1 of the first flip-flop **122** and appearing at the input terminal D2 of the second flip-flop **124** will be latched to the output terminal Q2 of the second flip-flop **124** as the data input is latched by flip-flop **122**. The output terminals Q1 and Q2 of the flip-flops **122**, **124** will now hold the new output values "a" and "b". This operation represents correct synchronous behavior.

The more realistic scenario, especially when gated clocks are used, is that the clock signal CLK will not reach both of the flip-flops **122** and **124** at the same instant in time. This realistic assumption is especially valid in the illustrated example in which the clock signal CLK must pass through the combinational logic **126** on the third configurable logic device **114** before it reaches the second flip-flop **124** on the fourth configurable logic device **116**. In this example, assume the clock signal CLK reaches the first flip-flop **122**

in the second configurable logic device **112** and clocks the value at that flip-flop's input terminal **D1** to the output **Q1**. At some point, the output **Q1** of the first flip-flop is now holding the new value "a" and this new value begins to propagate toward the input **D2** of the second flip-flop **124**. The rising edge of the clock signal CLK has not propagated to the second flip-flop **124** on the fourth configurable logic device **116**, however. Instead, a race of sorts is established between the rising edge of the clock signal CLK and the new value "a" to the second flip-flop **124**. If the new value "a" reaches the input terminal D2 of the second flip-flop before the rising edge of the clock signal CLK, the old value "b" will be over-written. This is incorrect behavior since the information contained in "b" is lost. For correct operation of the circuit, it was required that signal "b" at the input terminal **D2** of the second flip-flop **124** be held valid for a brief period of time after the arrival of the clock edge to satisfy a hold time requirement. Unfortunately, unpredictable routing and logic delays postpone the clock edge beyond the validity period for the input signal "b".

In environments where delays can not be predicted precisely, hold time violations are a serious problem that can not be rectified merely by stretching the length of the clock period. Often, there is a need for careful delay tuning in traditional systems, either manually or automatically, in which analog delays are added to signal paths in the logic. The delays usually require further decreases in the operational speed of the target system. This lengthens the periods of the environmental timing signals and gives the emulation system more time to perform the logic calculations. These changes, however, create their own timing problems, and further erode the overall speed, ease-of-use, and predictability of the system.

FIG. **4A** is a schematic diagram showing the internal architecture of the logic system **200** which has been configured according to the principles of the present invention. This logic system **200** comprises a plurality of configurable logic devices **214***a*–**214***d*. This, however, is not a strict necessity for the invention. Instead, the logic system **200** could also be constructed from a single logic device or alternatively from more than the four logic devices actually shown. The logic devices are shown as being connected by a Manhattan style interconnect **418**. Again, the interconnect is non-critical, modified Manhattan-style, crossbars or hierarchial interconnects are other possible and equivalent alternatives.

The internal logic architecture of each configurable logic device **214***a*–**214***d* comprises a finite state machine **428**–**434** and logic **420**–**426**. An internal or virtual clock VClk generates an internal or virtual clock signal that is distributed through the interconnect **418** to each logic device **214***a*–**214***d*, and specifically, the logic **420**–**426** and finite state machines **428**–**434**. Generally, the logic **420**–**426** performs the logic operations and state transitions associated with the logic design that was developed from the digital circuit description. The finite state machines **428**–**434** control the sequential operations of the logic in response to the signal from the virtual clock VClk.

The logic system **200** operates synchronously with the single internal clock signal VClk. Therefore, a first synchronizer SYNC1 and a second synchronizer SYNC2 are provided to essentially generate synchronous versions of timing signals from the environment. In the illustrated example, they receive environmental timing signals EClk1 and EClk2, respectively. The synchronizers SYNC1 and SYNC2 also receive the internal clock signal VClk. Each of the synchronizers SYNC1 and SYNC2 generates a synchronizing control signal V_{G01}, V_{G02} in response to an edge of the respective environmental timing signal EClk1 and EClk2, upon the next transition of the internal clock VClk. Thus, these control signals are synchronous with the internal clock.

FIG. **4B** shows an exemplary timing diagram of the virtual clock signal VClk compared with a first environmental clock signal EClk1 and a second environmental clock signal EClk2. As shown, typically, the virtual clock VClk is substantially faster than any of the environmental clocks, at least four times faster but usually faster by a factor of ten to twenty. As a general rule, the temporal resolution of the virtual clock, i.e., the cycle time or period of the virtual clock, should be smaller than the time difference between any pair of environmental timing signal edges.

In the example, the environmental clocks EClk1 and EClk2 are rising edge-active. The signals V_{G01} and V_{G02} from the first synchronizer SYNC1 and the second synchronizer SYNC2, respectively, are versions of the environmental clock which are synchronized to the internal clock VClk in duration. The transitions occur after the rising edges of the environmental clocks EClk1 and EClk2, upon the next or a later rising edge of the internal clock. For example, the second synchronizing signal V_{G02} is active in response to the receipt of the second environmental clock signal EClk2 upon the next rising edge of the internal clock VClk.

Returning to FIG. **4A**, in typical simulation or emulation configurable systems and the present invention, logic of the configurable devices include a number of interconnected combinational components that perform the boolean functions dictated by the digital circuit design. An example of such components are logic gates. Other logic is configured as sequential components. Sequential components have an output that is a function of the input and state and are clocked by a timing signal. An example of such sequential components would be a flip-flop. In the typical configurable systems, the environmental clock signals are provided to the logic in each configurable logic device to control sequential components in the logic. This architecture is a product of the emulated digital circuit design in which similar components were also clocked by these timing signals. The present invention, however, is configured so that each one of these sequential components in the logic sections **420**–**426** is clocked by the internal or virtual clock signal VClk. This control is schematically shown by the distribution of the internal clock signal VClk to each of the logic sections **420**–**426** of the configurable devices **410**–**416**. As described below, the internal clock is the sole clock applied to the sequential components in the logic sections **420**–**426** and this clock is preferably never gated.

Finite state machines **428**–**434** receive both the internal clock signal VClk and also the synchronizing signals V_{G01} V_{G02} from the synchronizers SYNC1 and SYNC2. The finite state machines **428**–**434** of each of the configurable logic devices **410**–**416** generate control signals to the logic sections **420**–**426**. These signals control the operation of the sequential logic components. Usually, the control signals are received at load enable terminals. As a result, the inherent functionality of the original digital circuit design is maintained. The sequential components of the logic are operated in response to environmental timing signals by virtue of the fact that loading occurs in response to the synchronized versions of the timing signals, i.e. V_{G01} V_{G02}. Synchronous operation is imposed, however, since the sequential components are actually clocked by the single internal clock signal VClk throughout the logic system **200**. In contrast, the typical simulation or emulation configurable systems would

clock the sequential components with the same environmental clock signals as in the original digital circuit description.

It should be noted that separate finite state machines are not required for each configurable logic device. Alternatively, a single finite state machine having the combined functionality of finite state machines **428–434** could be implemented. For example, one configurable device could be entirely dedicated to this combined finite state machine. Generally, however, at least one finite state machine on each device chip is preferred. The high cost of interconnect bandwidth compared to on-chip bandwidth makes it desirable to distribute only the synchronizing signals V_{G01} V_{G02} to each chip, and generate the multiple control signals on-chip to preserve the interconnect for other signal transmission.

FIG. **5**A shows a portion of a logic circuit that has been programmed into the logic system **200** according to the present invention. This logic circuit is a resynthesized version of the logic circuit shown in FIG. **3**. That is, the logic circuit of FIG. **5**A and of FIG. **3** have many of the same characteristics. Both comprise flip-flops **122** and **124**. The flip-flop **122** has an output terminal Q1 which connects to the input terminal D2 of flip-flop **124**. Further, the combinational logic **126** is found in both circuits.

The logic circuit of FIG. **5**A differs from FIG. **3** first in that each of the flip-flops **122** and **124** are load-enable type flip-flops and clocked by a single internal clock VClk. Also, the environmental clock signal Clk is not distributed per se to both of the flip-flops **122** and **124** as in the circuit of FIG. **3**. Instead, a synchronized version of the clock signal V_{G0} is distributed to a finite state machine **430** of the second configurable logic device **214***b* and is also distributed to a finite state machine **434** of the fourth configurable logic device **214***d*. The finite state machine **430** then provides a control signal to a load enable terminal LE1 of flip-flop **122** and finite state machine **434** provides a control signal to the load enable terminal LE2 of flip-flop **124** through the combinational logic **126**.

FIG. **5**B is a timing diagram showing the timing of the signals in the circuit of FIG. **5**A. That is, at time **510**, new data is provided at the input terminal D1 of flip-flop **522**. Then, at some later time, **512**, the clock signal Clk is provided to enable the flip-flop **122** to clock in this new data. The second flip-flop **124** is also intended to respond to the environmental clock signal Clk by capturing the previous output of flip-flop **122** before that flip-flop is updated with the new data. Recall that the problem in the logic circuit of FIG. **3** was that the clock signal to the second flip-flop **124** was gated by the combinational logic **126** which delayed that clock signal beyond time at which the output "b" from the output terminal Q1 of the flip-flop **122** was valid. In the present invention, the environmental clock signal Clk is received at the synchronizer SYNC. This synchronizer also receives the virtual clock signal VClk. The output of the synchronizer V_{G0} is essentially the version of the environmental clock signal that is synchronized to the internal clock signal. Specifically, the new signal V_{G0} has rising and falling edges that correspond to the rising edges of the internal clock signal VClk.

The finite state machines **430** and **434** are individually designed to control the flip-flops in the respective configurable logic **214***b* and **214***d* to function as required for correct synchronous operation. Specifically, finite state machine **434** generates a control signal **215** which propagates through the combinational logic **126** to the load enable terminal LE2 of the flip-flop **124**. This propagation of control signal **215** from finite state machine, through combinational logic **126**, to LE2 occurs in a single virtual clock cycle. The generation of control signal **215** precedes the generation of control signal **217** by the finite state machine **430** by a time of two periods (for example) of the internal clock VClk. This two cycle difference, **514**, assumes that flip-flop **124** is enabled before flip-flop **122** is enabled, thereby latching "b", and thus providing correct operation. As a result, both flip-flop **122** and flip-flop **124** are load enabled in a sequence that guarantees that a new value in flip-flop **122** does not reach flip-flop **124** before flip-flop **124** is enabled. In fact, if the compiler has scheduled "b" to arrive at D2 on some cycle, x, later than **217**, then the compiler can cause control signal **215** to be available on that cycle x, or later. In the above instance, the correct circuit semantics is preserved even though control signal **215** arrives after control signal **217**. The key is that **215** must enable flip-flop **124** in a virtual cycle in which "b" is at D2.

Further, the precise control of storage elements afforded by the present invention allows set up and hold times into the target system to be dictated. In FIG. **5**A, output Q2 of flip-flop **124** is linked to a target system via a third flip-flop **140**. The flip-flop **140** is load enabled under the control of finite state machine **434** and clocked by the virtual clock. Thus, by properly constructing this finite state machine **434**, the time for which flip-flop **140** holds a value at terminal Q3 is controllable to the temporal resolution of a cycle or period of the virtual clock signal.

This aspect of the invention enables the user to test best case and worst case situations for signal transmission to the target system and thereby ensure that the target system properly captures these signals. In a similar vein, this control also allows the user to control the precise time of sampling signals from the target system by properly connected storage devices.

FIG. **6** illustrates a method by which a digital circuit design with an arbitrary clocking methodology and state elements is transformed into a new circuit that is synchronous with the internal clock signal but is a functional equivalent of the original digital circuit. The state elements of the new circuit are exclusively edge triggered flip-flops.

The first step is specification **610**. This is a process by which the digital circuit design along with all of the inherent timing methodology information required to precisely define the circuit functionality is identified. This information is expressed in four pieces, a first piece of which is the gate-level circuit netlist **610***a*. This specifies the components from which the digital circuit is constructed and the precise interconnectivity of the components.

The second part **610***b* of the specification step **610** is the generation of a functional description of each component in the digital circuit at the logic level. For combinatorial components, this is a specification of each output as a boolean function of one or more inputs. For example, the specification of a three input OR gate—inputs A, B, and C and an output O—is O=A+B+C. For sequential components, this entails the specification of outputs as a boolean function of the inputs and state. The specification of the new state as a boolean function of the inputs and state is also required for the sequential components along with the specification of when state transitions occur as a function of either boolean inputs or directed input transitions. A directed input transition is a rising or falling edge of an input signal, usually a timing signal from the environment in which the logic system **200** is intended to ultimately function. For example, the specification of a rising edge-triggered flip-flop—inputs

D, CLK, of output Q, and state S— is Q=S, S=D, and state transition when CLK rises.

Another part of the specification step is the description of the timing relationships of the inputs to the logic system step **610***c*. This includes environment timing signals and environmental input signals and the relationship to the output signals generated by the logic system **200** to the environment. Input signals to the logic system **200** can be divided into two classes: timing signals and environmental data signals. The timing signals are generally environmental clock signals, but can also be asynchronous resets and any other form of asynchronous signal that combinatorially reach inputs of state elements involved in the functions triggering state transitions. In contrast, environmental data signals include environmental output signals and output signals to the environment that do not combinatorially reach transition controlling inputs of state elements. The timing relationship also specify the timing of environmental data signals relative to a timing signal.

The specification step must also include the specification of the relative timing relationships for all timing signals step **610***d*. These relationships can be one of three types:

A basket of timing signals can be phase-locked. Two signals of equal frequency are phase-locked if there is a known phase relationship between each edge of one signal and each edge of the other signal. For example, the first environmental clock signal and the second environmental clock signal illustrated in FIG. **4** would be phase-locked signals. Additionally, two signals of integrally related frequency are phase-locked if there is a known phase relationship, relative to the slower signal, between any edge of the slower signal and each edge of the faster signal. Two signals of rationally related frequency are phase-locked if they each are phase-locked to the same slower signal.

Another type of timing relationship is non-simultaneous. Two signals are non-simultaneous if a directed transition in one signal guarantees that no directed transition will occur in the other within a window around the transition of some specified finite duration. If two signals are non-simultaneous and also not phase-locked, this implies that one signal is turned off while the other is on and vice versa. For example, two non-simultaneous signals might be two signals that indicate the mutually exclusive state of some component in the environment. The first signal would indicate if the component was in a first condition and the second timing signal would indicate if the component were in a second condition and the first and second condition could never happen at the same time.

Finally, the last type of relationship is asynchronous. Two signals are asynchronous if the knowledge about a directed transition of one of the signals imparts no information as to occurrence of a transition in the other signal.

It should be recognized that phase-locked is a transitive relationship so that there will be collections of one or more clocks that are mutually phase-locked with respect to each other. Such collection of phase-locked clocks is referred to as a domain. Relationship between domains are either non-simultaneous or asynchronous. The timing signals must be decomposed into a collection of phase-locked domains, and the relationship between pairs of the resulting domains, either synchronous or non-simultaneous, must be specified.

The ordering of the edges of timing signals within each domain are also specified. For example, first CLK1 rises, then CLK2 rises, then CLK2 falls and then CLK1 falls.

A transition analysis step **612**, value analysis step **614**, and sampling analysis step **616** are used to determine when, relative to the times at which transitions occur on timing signals, signals within a digital circuit change value, and where possible, what these values are. Also determined is when the values of particular signals are sampled by state elements within the circuit as a separate analysis.

In the transition analysis step **612**, a discrete time range is established for each clock domain including one time point for each edge of a clock within the domain. All edges within the domain are ordered and the ordering of time points corresponds to this ordering of edges.

In the value analysis step **614**, the steady state characteristics of every wire in the digital circuit is determined for each discrete time range. Within a discrete time range, any wire within the digital circuit can either be known to be O, known to be 1, known not to rise, known not to fall or known not to change, or a combination of not falling and not rising. A conservative estimate of the behavior of an output of a logic component can be deduced from the behavior of its inputs. Information about environmental timing signals and environmental data signals can be used to define their behavior. Based on the transition and value information of the inputs to the logic system corresponding information can be deduced for the outputs of each component. A relaxation algorithm is used, in which output values of a given component are recomputed any time an input changes. If the outputs in turn change, this information is propagated to all the places the output connects, since these represent more inputs which have changed. The process continues until no further changes occur.

A second relaxation process, similar to that for transition and value analysis, is used in the sampling step **616**. Sampling information reflects the fact that at some point in time, the value carried on a wire may be sampled by a state element, either within the logic system **200** or by the environment. Timing information for output data signals to the environment provides an external boundary condition for this relaxation process. Additionally, once transition analysis has occurred, it is possible to characterize when all internal state elements potentially make transitions and thus when they may sample internal wires. Just as with transition and value propagation, the result is a relationship between inputs and outputs of a component. For sampling analysis, it is possible to deduce the sampling behavior of inputs of a component from the sampling information for its outputs. The relaxation process for computing sampling information thus propagates in the opposite direction from that of transition information, but otherwise similarly starts with boundary information and propagates changes until no further changes occur.

At the termination of transition **612** and sampling **616** steps it is possible to characterize precisely which timing edges can result in transitions and/or sampling for each wire within the digital circuit. Signals which are combinationally derived from timing signals with known values often also carry knowledge about their precise values during some or all of the discrete time range. They similarly often are known to only be able to make one form of directed transition, either rising or falling, at some particular discrete time point. This information is relevant to understanding the behavior of edge-triggered state elements.

The final resynthesis step **618** involves the application of a number of circuit transformations to the original digital circuit design which have a number of effects. First, the internal clock VClk is introduced into the logic design **200** of the digital circuit. The internal clock signal is the main clock of the logic system **200**. Further, in effect, all of the

original environmental timing signals of the digital circuit are converted into data signals in the logic system **200**. Finally, all of the state elements in the digital circuit are converted to use the internal clock signal as their clock, leaving the internal clock as the only clock signal of the transformed system. The state elements of the original digital circuit design are converted preferably into edge-triggered flip-flops and finite state machines, which generate control signals to the load enable terminals of the flip-flops. The information developed in the transition analysis step **612**, value analysis step **614**, and sampling analysis step **616** is used to define the operation of the finite state machines as it relates to the control of the flip-flops in response to the internal clock signal and the environmental timing signals. The finite state machines send load enable signals to the flip-flops when it is known that data inputs are correct based upon a routing and scheduling algorithm described in the U.S. patent application Ser. No. 08/344,723 filed Nov. 23, 1994 and entitled "Pipe-Lined Static Router and Scheduler for Configurable Logic System Performing Simultaneous Communications and Computations", incorporated herein by this reference. The scheduling algorithm essentially produces a load enable signal on a virtual clock cycle that is given by the maximum of the sum of data, value available time, and routing delays for each signal that can affect data input.

Single Flip-Flop Timing Resynthesis

FIG. **7A** shows a simple edge-triggered flip-flop **710** which was a state element in the original digital circuit. Specifically, the edge-triggered flip-flop **710** receives some input signal at its input terminal D and some timing signal, such as an environmental clock signal ECLK at its clock input terminal. In response to a rising edge received into this clock terminal, the value held at the input terminal D is placed at the output terminal Q.

The timing resynthesis step converts this simple edge-triggered flip-flop **710** to the circuit shown in FIG. **7B**. The new flip-flop is a load-enabled flip-flop and is clocked by the internal clock signal VClk. The enable signal of the converted flip-flop is generated by a finite state machine FSM. Specifically, the finite state machine monitors a synchronized version of the clock signal V_{G0} and asserts the enable signal to the enable input terminal E of the converted flip-flop **720** for exactly one cycle of the internal clock VClk in response to synchronizing signal V_{G0} transitions from 0 to 1. The finite state machine is programmed so that the enable signal is asserted on an internal clock signal cycle when the input IN is valid accounting for delays in the circuit that arise out of a need to route the signal IN on several VClk cycles from the place it is generated to its destination at the input of flip-flop **720**. In a virtual wire systems signals are routed among multiple FPGAs on specific internal clock VClk cycles. The synchronizing signal V_{G0} is generated by a synchronizer SYNC in response to receiving the environmental timing signal EClk on the next or a following transition of the internal clock signal VClk. As a result, the circuit is functionally equivalent to the original circuit shown in FIG. **7A** since the generation of the enable signal occurs in response to the environmental clock signal EClk each time a transition occurs. The circuit, however, is synchronous with the internal clock VClk.

In a digital circuit comprising combinational logic and a collection of flip-flops, all of which trigger off the same edge of a single clock, the basic timing resynthesis transformation, shown in FIG. **7B** and described above, can be extended. All flip-flops are converted to load-enabled flip-flops and have their clock inputs connected to the internal clock VClk. The load enable terminal E of each flip-flop is connected to enable signals generated by a shared finite state machine in an identical manner as illustrated above. The FSM can be distinct for each FPGA. The enables for each flip-flop will be produced to account for routing delays associated with each signal input to the flip-flops.

Timing Resynthesis for Domains for Multiple Clocks

FIG. **8A** shows a circuit comprising three flip-flops **810–814** that are clocked by two environmental clock signals EClk**0** and EClk**1**. For the purposes of this description, both environmental clock signals EClk**0** and EClk**1** are assumed to be phase-locked with respect to each other.

The transformed circuit is shown in FIG. **8B**. It should be noted that the basic methodology of the transform is the same as described in relation to FIGS. **7A** and **7B**. The finite state machine FSM and the clock sampling circuitry SYNC1 and SYNC2 have been extended. As before, each flip-flop of the transformed circuit has been replaced with a load-enabled positive-edge triggered flip-flop **820–824** in the transformed circuit. The first environmental clock signal EClk**0** and the second environmental clock signal EClk**1** are synchronized to the internal clock by the first synchronizer SYNC**0** and the second synchronizer SYNC**1**. The synchronizing signals V_{G00} and V_{G01} are generated by the synchronizers SYNC**0** and SYNC**1** to the finite state machine FSM. The finite state machine FSM watches for the synchronizing signals V_{G00} and V_{G01} and then produces a distinct load enable pulse C**0**-Rise, C**0**-Fall, C**1**-Rise for each timing edge on which the clocks EClk**0** and EClk**1** of the flip-flops **820–824** operate. The ordering of these load enable pulses is prespecified within a domain where there is a unique ordering of the edges of all phase-locked clocks. This unique ordering of clocks is specified by the user of the system. As with the single clock case shown in FIG. **7B**, each of the enable pulses C**0**-Rise, C**0**-Fall, and C**1**-Rise is asserted for exactly one period of the internal clock VClk upon detection of the corresponding clock edge in FIG. **8B**.

Multiple Clock Domains Resynthesis

FIG. **9A** shows a collection of flip-flops **910–912** from the digital circuit having multiple clock domains. That is, the first clock signal CLK**0** and the second clock signal CLK**1** do not have a phase-locked relationship to each other, rather the clocks are asynchronous with respect to each other.

FIG. **9B** shows the transformed circuit. A different finite state machine FSM**0** and FSM**1** is assigned to each domain. Specifically, a first finite state machine FSM**0** is synchronized to the first environmental clock EClk**0** to generate the load enable signal to the load enable terminal E**0** of the first flip-flop **920**. The second finite state machine FSM**1** generates a load enable signal to E**1** of the second flip-flop **922** in response to the second environmental clock signal EClk**1**. It should be noted, however, that although FSM**0** and FSM**1** operate independently of each other, each of whose sequences are initiated by separate signals V_{g00} and V_{g01}, and that although the first flip-flop **920** and the second flip-flop **922** work independently of each other, i.e., load enabled by different clock signals EClk**0** and EClk**1**, the resulting system is a single-clock synchronous system with the internal clock VClk.

The relationship between the behavior of the first finite state machine FSM**0** and the second finite state machine FSM**1** of the two clock signal domains is related to the relationship between the domains themselves. When the two domains are asynchronous, the first finite state machine FSM**0** and the second finite state machine FSM**1** may operate simultaneously or non-simultaneously. When the

two domains are non-overlapping, the first finite state machine FSM0 and the second finite state machine FSM1 never operate simultaneously since the edges within the domains are separated in time.

The simultaneity of operation of finite state machines that are asynchronous with respect to each other leaves two circuits which can not readily be transformed by timing resynthesis. A state element which can undergo transitions as a result of an edge produced from a combination of signals in asynchronously related domains can not be resynthesized. Such condition can rise if two asynchronous clocks are gated together and fed into the clock input of a flip-flop or if a state element with multiple clocks and/or asynchronous presets or clears is used as transition triggering inputs from distinct asynchronously related clock domains. Due to the non-simultaneous events and non-overlapping domains, the situations above are not problematic in the non-overlapping situation.

Gated Clock Transformations

Clock gating in the digital circuit provides additional control over the behavior of state elements by using combinational logic to compute the input to clock terminals. The timing resynthesis process transforms gated clock structures into functionally equivalent circuitry which has no clock gating. Generally, gated clock structures can be divided into two classes: simple gated clocks and complex gated clocks. The basis for this distinction lies in the behavior of the gated clocks as deduced from timing analysis. Previously, the terms timing signal and data signal were defined in the context of inputs and outputs to the digital circuit. A gated clock is a combinational function of both timing signals and data signals. The gated clock transition then controls the input of a state element. Data signals can either be external input data signals from the environment or internally generated data signals.

A simple gated clock has two properties:

1) at any discrete time it is possible for a simple gated clock to make a transition in at most one direction, stated differently, there is no discrete time at which the simple gated clock may sometime rise and sometime fall; and
2) only timing signals change at those discrete times at which state elements can change state.

A complex gated clock violates one of these two properties.

Simple Gated Clock Transformation

FIG. **10**A shows a circuit that exhibits a simple gated clock behavior. FIG. **10**B is a timing diagram showing transitions in the data signal and the gated clock signal as a function of the environmental clock signal EClk. Specifically, upon the falling edge of the environmental clock EClk, the gating flip-flop **1010** latches the control signal CTL received at its input D1 at its output terminal Q1. This is the data signal. The AND gate **1012** receives both the data signal and the environmental clock EClk. As a result, only when the environmental clock EClk goes high, does the gated-clock signal go high on the assumption that the data signal is also a logic high. Upon the rising edge of the gated clock, the second flip-flop **1014** places the input signal IN received at its D2 terminal to its output terminal Q2.

FIG. **10**C shows the transformed circuit. Here, a finite state machine FSM receives a signal V_{GO} from the synchronizer SYNC upon receipt of the environmental clock EClk. The finite state machine FSM produces two output signals: C0-Fall which is active upon the falling edge of the environmental clock signal, and C0-Rise which is active upon the rising of the environmental clock signal EClk.

The transformed circuit functions as follows. On the first period of the internal clock VClk after the falling edge of the environmental clock signal EClk, the first flip-flop **1016** places the value of the control signal received at its input terminal D1 to its output terminal Q1 upon the clocking of the internal clock signal VClk. This output of the first flip-flop **1016** appearing at terminal Q1 corresponds to the data signal in the original circuit. This data signal is then combined in an AND gate **1020** with the signal C0-Rise from the finite state machine FSM that is indicative of the rising edge of the environmental clock signal EClk. The output of the AND gate goes to the load enable terminal E2 of a second flip-flop **1018** which receives signal IN at its input terminal D2. Again, upon the receipt of this load enable and upon the next cycle of the internal clock VClk, the second flip-flop moves the value at its input terminal D2 to its output terminal Q2.

Complex Gated Clock Transformations

In the case of complex gated clock behavior, the factoring technique used for simple gated clock transformations is inadequate. Because data and clocks change simultaneously and/or the direction of a transition is not guaranteed, both the value of a gated clock prior to the transition time and the value of the gated clock after the transition time are needed. Using these two values, it can be determined whether a signal transition that should trigger a state change has occurred. One way to produce the post-transition value of data signals is to replicate the logic computing the signal and also replicate any flip-flops containing values from which the signal is computed and which may change state as a result of the transition. These replica flip-flops can be enabled with an early version of the control signal, thus causing them to take on a new state prior to the main transition. By this mechanism, pre- and post-transition values for signals needed for gated clocks can be produced.

An alternative way to get the two required values for the gated clock signal is to add a flip-flop to record the pre-transition state of the gated-clock and delay in time the update of the state element dependent on the gated clock. These two techniques have different overhead costs and the latter is only applicable if the output of the state element receiving the gated clock is not sampled at the time of the transition. The former always works but the latter generally has lower overhead when applicable.

FIG. **11**A shows two cascaded edge-triggered flip-flops **1100** and **1112**. This configuration is generally known as a frequency divider. The environmental clock signal EClk is received at the clock terminal of the first flip-flop **1110**; and at the output Q2 of the second flip-flop **1112**, a new clock signal is generated that has one-fourth the frequency of EClk. The divider of FIG. **11**A operates as follows: In an initial state in which the output terminal Q1 of the first flip-flop **1110** is a 0 and the input terminal D1 of the flip-flop **1110** is a 1, receipt of the rising edge of the environmental timing signal EClk changes Q1 to a 1 and D1 converts to a 0. The conversion of Q1 from 0 to 1 functions as a gated clock to the clock input terminal of the second flip-flop **1112**. The second flip-flop **1112** functions similarly, but since it is only clocked when Q1 of the first flip-flop **1110** changes from 0 to 1, but not 1 to 0, it changes with one-fourth the frequency of EClk.

FIG. **11**B shows the transformed circuit of FIG. **11**A. Here, a replica flip-flop **1120** has been added that essentially mimics the operation of the first flip-flop **1122**. The replica flip-flop **1120**, however, receives a pre-Clk-Rise control signal from the finite state machine FSM. More specifically, the finite state machine FSM responds to the synchronizing signal V_{GO} and the internal clock VClk and produces a pre-CLK-rise signal that is active just prior to the CLK-Rise

signal, CLK-Rise being active in response to the rising edge of the environmental timing signal EClk. Assume the output terminal Q1 of the first flip-flop **1122** is initially at a 0 and the input terminal D1 of first flip-flop **1122** is a 1, the replica flip-flop **1120** is initially at a 0. Upon receipt of the pre-CLK-rise signal at the replica flip-flop load enable terminal ER, the output terminal QR of the replica flip-flop **1120** makes a transition from a 0 to a 1. Since Q1 is low and QR is high, an AND gate **1124** functioning as an edge detector generates a high signal. When the CLK-rise control signal from the finite state machine FSM is active in response to receipt of the rising edge of the environmental clock signal EClk, the output terminal Q1 of the first flip-flop **1122** is converted from a 0 to a 1. The enable terminal E2 of flip-flop **1126** also is high, causing the flip-flop to change state. On the next falling transition of Q1, the AND gate **1124** will produce 0 and flip-flop **1126** will not change state. Since the replica flip-flop **1120** provides a zero to the rising edge detector whenever the zero is present at the input terminal of the first flip-flop, the rising edge detector is enabled only every other transition of Q1.

Latch Resynthesis

Generally, latches are distinguished from flip-flops in that flip-flops are edge-triggered. That is, in response to receiving either a rising or falling edge of a clock signal, the flip-flop changes state. In contradistinction, a latch has two states. In an open state, the input signal received at a D terminal is simply transferred to an output terminal Q. In short, in an open condition, the output follows the input like a simple wire. When the latch is closed, the state of the output terminal Q is maintained or held independent of the input value at terminal D. A semantic characterization of such a latch is as follows. For an input D, output Q, a gate G, and a state S, Q=S. S=D if G=1. The latch is open when G=1 and closed when G=0.

Beginning with the simplest case, if the output of a latch is never sampled when the latch is closed, G=0, the latch is really just a wire. Latches with this characteristic may be used to provide extra hold time for a signal. For this sample latch, this would be true, if the set of discrete times at which the output of the latch is sampled, is equal to or a proper subset of the set of discrete times at which the gate signal G is known to have a value of 1. In this situation, the latch can be removed and replaced with a wire connecting the input and output signals.

In contrast, if the output of the latch is never sampled when the latch is open, the latch is equivalent to a flip-flop. The only value produced by the latch which is ever sampled is a value of the input D on the gate signal edge when the latch transitions from open to closed. This condition is true if the set of discrete times at which the output of the latch is sampled, is equal to, or a proper subset of the discrete times at which the gate signal G is known to have a value of 0. In this situation, the latch can be removed and replaced with an edge-triggered flip-flop.

As shown in FIG. **12**, latches that are open when their gate signal G is high **1210** are converted to negative-edge triggered flip-flops **1212**. Latches that are open when their gate signal G is low **1214** are converted to positive edge triggered flip-flops **1216**.

Once the transition from the latch to the edge triggered flip-flop has been made, these new edge-triggered flip-flops are then further resynthesized by the timing resynthesis techniques described in connection with FIGS. **6–11**. Therefore, after this further processing, both positive and negative edge-triggered flip-flops will be flip-flops clocked by the internal clock VClk. The resynthesized flip-flops will have an enable signal that is generated by a finite state machine in response to the particular environmental clock signal that gated the original latch element.

Referring to FIG. **13**, in the condition in which the output of a given latch **1310** is sampled both when the latch might be open and might be closed, that latch can be converted to a flip-flop **1312**, plus a multiplexor **1314** as shown in FIG. **13**. There, when the gate signal G is low, the multiplexor **1314** selects the input signal to the input terminal D of the flip-flop **1312**. On the rising edge of the gating signal, however, the input to the D terminal is latched at the output terminal Q. Also, at this point, the gating signal selects the second input to the multiplexor **1214**. As with the case in FIG. **12**, the result of the transformation in FIG. **13** is subjected to further resynthesis.

The transform of FIG. **13** may exhibit timing problems if the multiplexor is implemented in a technology that exhibits hazards, or output glitches. Output glitches can and could result in set up and hold time problems of the sampling state element. This transformation can therefore only be used when the output is never sampled at discrete times at which the clock may exhibit an edge. If the output is sampled both when the latch might be opened and closed and some sampling occurs on the edge of the gate signal, a final transformation is employed. A new clock signal is created which is phase-locked to the original clock signal and precedes it.

As shown in FIGS. **14A** and **14B**, the latch **1410** of FIG. **14A** is replaced by a flip-flop which receives the phase-advanced clock indicated by the negative delay **1412** as shown in FIG. **14B**. The state transition of the new flip-flop **1414** precedes a state transition of any circuits sampling the original output Q of the original latch **1410**. If the latch is also sampled when it is open by signals occurring prior to the sampling edge, one of the prior techniques can be employed, either latch to wire or latch to flip-flop and multiplexor transforms of FIG. **13**.

As shown in FIG. **14B**, the negative delay **1412** represents a time-advanced copy of the clock CLK which is used to clock the flip-flop **1414**. While negative-delays are unphysical, this structure can be processed by the timing resynthesis process with a distinct control signal generated by a finite state machine.

FIG. **15** shows a finite state machine FSM generating a pre-CLK-rise control signal one or more cycles of the internal clock VClk prior to the generation of the control signal, CLK-Rise. The control signal CLK-rise is generated in response to the rising edge of the environmental timing signal EClk. As a result, the input signal appearing at the D terminal of the flip-flop **1510** is transferred to the output terminal prior to the rising edge of the environmental clock signal EClk as signaled by the Clk-rise control signal. Subsequent elements can be then load enabled from the CLK-Rise signal generated by the finite state machine FSM. Here again, if the latch of the original digital circuit is sampled both when the latch is opened and closed, a multiplexor can be placed at the output Q of the flip-flop **1510**.

Combinational Loop Transformations

Combinational loops with an even number of logic inversions around the loop are an implicit state element. An example is shown in FIG. **16A**, this implicit state can be transformed into an explicit state element which is clocked by the virtual clock VClk by simply choosing a wire **1601** in the loop and inserting a flip-flop **1602** which is clocked by the virtual clock VClk as shown in FIG. **16B**.

The addition of the flip-flop **2602** changes the timing characteristics of the loop. Additional virtual clock cycles are required for the values in the loop to settle into their final states.

Assume in FIG. **16B** that all input values to the loop are ready by some virtual cycle V. In the absence of the flip-flop **1602**, all outputs will become correct and stable after some delay period. With the flip-flop **1602**, it is necessary to wait until the loop stabilizes and then wait for an additional virtual clock period during which the flip-flop value may change and subsequently change the loop outputs. Thus the outputs of the loop cannot be sampled until virtual cycle V+1.

If combinational cycles are nested, each can be broken by the insertion of a flip-flop as above. Nested loops may require up to 2^N clock cycles to settle, where N is the depth of the loop nesting and thus the number of flip-flops needed to break all loops.

RS Latch Transformations

RS latches **1710** are asynchronous state elements built from cross-coupled NOR or NAND gates **1712**, as illustrated in FIG. **17A**.

RS latches **1710** can be transformed based on the transformation for combinational cycles illustrated in FIGS. **16A** and **16B**. An alternative approach illustrated in FIG. **17B** eliminates the combinational cycles associated with RS latches while also avoiding the extended settling time associated with the general combinational cycle transformation of FIG. **16B**.

The circuit is FIG. **17B** forces the outputs Q and $\overline{Q}$ of the RS latch **1710** combinatorially to their values for all input patterns except the one in which the latch maintains its state. For this pattern, the added flip-flop **1714** produces appropriate values on the outputs. Logic **1716** is provided to set the flip-flop **1714** into an appropriate state, based on the values of the inputs whenever an input pattern dictates a state change. When the latch **1710** is maintaining its state, the outputs will be stable so no propagation is required. Thus the outputs of the transformation are available with only a combinatorial delay.

A symmetrical transformation can be applied to latches produced from cross-coupled NOR gates.

Asynchronous Presets and Clears

Asynchronous presets and clears of state elements shown in FIG. **18A** can be transformed in one of two ways. Each transformation relies on the fact that preset and clear signals R are always synchronized to the virtual clock, either because they are internally generated by circuitry which is transformed to be synchronous to the virtual clock or because they are external asynchronous signals which are explicitly synchronized using synchronizer circuitry.

The first transformation, shown in FIG. **18B**, makes use of an asynchronous preset or clear on flip-flop **1808** in the FPGA, if such exists. The enable signal E which enables the resynthesized state element to undergo state changes is used to suppress/defer transitions on the preset or clear input R to eliminate race conditions arising from simultaneously clocking and clearing or presenting a state element.

The second transformation shown in FIG. **18C** converts an asynchronous preset or clear R_V which has already been synchronized to the clock into a synchronous preset or clear. The enable signal E to the resynthesized state element must be modified to be enabled at any time at which a preset or clear transition might occur by gate **1810**.

Returning to FIG. **6**, the above described transformations of the timing resynthesis step **618** in combination of with the specification step **610**, transition analysis **612**, value analysis **614** and sampling analysis **616** enable conversion of a digital circuit description having some arbitrary clocking methodology to a single clock synchronous circuit. The result is a circuit which the state elements are edge-triggered flip-flops.

To generate the logic system **200** having the internal architecture shown in FIG. **4**, this resynthesized circuit must now be compiled for and loaded into the configurable logic devices **410–416** by the host workstation **222**.

FIG. **19** shows the complete compilation process performed by the host workstation **222** to translate the digital circuit description into the configuration data received by the configurable devices **214**. More specifically, the input to a compiler running on the host workstation **222** is the digital circuit description in step **1610**. This description is used to generate the resynthesized circuit as described above. The result is a logic netlist of the resynthesized circuit **1611**. This includes the new circuit elements and the new VClk.

In step **1612**, functional simulations of the transformed circuit can be performed. This step ensures that the resynthesized circuit netlist is the functional equivalent of the original digital circuit. It should be noted that the transformed circuit is also more amenable to computer-based simulations. All relevant timing information specifying the behavior of the timing signals including the timing relationship to each other is built into the resynthesized circuit yet the resynthesized circuit is synchronous with a single clock. Therefore, the resynthesized circuit could alternatively be used as the circuit specification for a computer simulation rather than the hardware based simulation on the configurable logic system. The resynthesized circuit is then partitioned **1613** into the logic partition blocks that can fit into the individual FPGAs of the array, see FIG. **2**.

In the preferred embodiment of the present invention, techniques described in U.S. patent application Ser. No. 08/042,151, filed on Apr. 2, 1993, entitled Virtual Wires for Reconfigurable Logic System, which is incorporated herein by this reference, are implemented to better utilize pin resources by multiplexing global link transmission on the pins of the FPGAs across the interconnect. Additionally, as described in incorporated U.S. patent application Ser. No. 08/344,723, filed on Nov. 23, 1994, entitled Pipe-Lined Static Router and Scheduler for Configurable Logic System Performing Simultaneous Communication and Computation, signal routing is scheduled so that logic computation and global link transmission through the interconnect happen simultaneously.

Specifically, because a combinatorial signal may pass through several FPGA partitions as global links during an emulated clock cycle, all signals will not be ready to schedule at the same time. This is best solved by performing a dependency analysis, step **1614** on global links that leave a logic partition block. To determine dependencies, the partition circuit is analyzed by backtracing from partition outputs, either output global links or output signals to the target system, to determine on which partition inputs, either input links or input signals from the target system, the outputs depend. In backtracing, it is assumed that all outputs depend on all inputs for gate library parts, and no outputs depend on any inputs for latch or register library parts. If there are no combinatorial loops that cross partition boundaries, this analysis produces a directed acyclic graph, used by a global router. If there are combinatorial loops, then the loops can be hardwired or implemented in a single FPGA. Loops can also be broken by inserting a flip-flop into the loop and allowing enough virtual cycles for signal values to settle to a stable state in the flip-flop.

Individual FPGA partitions must be placed into specific FPGAs (step **1616**). An ideal placement minimizes system communication, requiring fewer virtual wire cycles to transfer information. A preferred embodiment first makes a random placement followed by cost-reduction swaps and then optimizes with simulated annealing. During global

routing (step **1618**), each global link is scheduled to be transferred across the interconnect during a particular period of the pipe-line clock. This step is discussed more completely in the incorporated U.S. patent application Ser. No. 08/344,723, Pipe-Lined Static Router and Scheduler for Configurable Logic System Performing Simultaneous Communication and Computation.

Once global routing is completed, appropriately-sized multiplexors or shift loops, pipeline registers, and associated logic such as the finite state machines that control both the design circuit elements and the multiplexors and pipeline registers are added to each partition to complete the internal configuration of each FPGA chip **22** (steps **1620**). See specifically, incorporated U.S. patent application Ser. No. 08/042,151, Virtual Wires for Reconfigurable Logic System. At this point, there is one netlist for each configurable logic device **214** or FPGA chip. These FPGA netlists are then processed in the vender-specific FPGA place-and-route software (step **1622**) to produce configuration bit streams (step **1624**). Technically, there is no additional hardware support for the multiplexing logic which time-multiplex the global links through the interconnect: the array of configurable logic is itself configured to provide the support. The necessary "hardware" is compiled directly into the configuration of the FPGA chip **214**. Some hardware support in the form of special logic for synchronizers to synchronize the external clocks to the internal VClk is recommended.

While this invention has been particularly shown and describe with references to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the spirit and scope of the invention as defined by the appended claims. For example, it is not a strict necessity that the internal clock signal VClk be distributed directly to the sequential logic elements. Preferably it reaches each element at substantially the same time. In some larger networks, therefore, some delay may be preferable to delay tune the circuit for propagation delays.

We claim:

1. A method of configuring a configurable logic system to operate in an environment, the logic system generating output signals to the environment in response to at least one environmental timing signal and environmental data signals provided from the environment, the method comprising:

defining an internal clock signal;

configuring the logic system to perform logic operations for generating the output signals in response to the environmental data signals and the internal clock signal; and

configuring the logic system to have a controller for coordinating operation of the logic operations in response to the internal clock signal and the environmental timing signal.

2. A method of configuring as described in claim **1**, further comprising configuring the controller to comprise a synchronizer for sampling the environmental timing signal in response to the internal clock signal.

3. A method of configuring as described in claim **2**, further comprising configuring the controller to further comprise a finite state machine for generating control signals to control the logic operations in response to the sampled environmental timing signal.

4. A method of configuring as described in claim **1**, further comprising configuring the logic system to have combinational logic and sequential logic to perform the logic operations.

5. A method of configuring as described in claim **4**, further comprising configuring the controller to comprise a finite state machine for generating control signals to the sequential logic in response to the environmental timing signal and the internal clock signal.

6. A method of configuring as described in claim **5**, further comprising configuring the sequential logic to comprise flip-flops receiving the internal clock signal at a clock input and the control signals at a latch enable input.

7. A method of configuring as described in claim **1**, wherein the logic system comprises at least one field programmable gate array.

8. A method of configuring as described in claim **1**, wherein the logic system comprises a plurality of configurable logic devices electrically connected via an interconnect for transmitting signals between the chips.

9. A method of configuring as described in claim **8**, wherein the interconnect comprises cross bar chips.

10. A method as configuring as described in claim **8**, wherein the interconnect utilizes a direct-connect topology.

11. A method of configuring as described in claim **10**, wherein the interconnect includes buses.

12. A method of configuring as described in claim **1**, further comprising configuring the controller to dictate set-up and hold times of signals to the environment.

13. A method of configuring as described in claim **1**, further comprising configuring the controller to dictate sampling times of the environmental data signals.

14. A method for converting a digital circuit design into a new circuit that is substantially functionally equivalent to the digital circuit design, the digital circuit design and the new circuit being adapted to operate in an environment in response to at least one environmental timing signal and environmental data signals and providing output data signals to the environment, the method comprising:

defining an internal clock signal; and

resynthesizing sequential logic elements in the digital circuit design that are clocked by the environmental timing signal to sequential logic elements in the new circuit that are clocked by the internal clock signal.

15. A method as claimed in claim **14**, wherein the resynthesized sequential logic elements of the new circuit are load enabled in response to the environmental timing signal.

16. A method as claimed in claim **14**, wherein the internal clock signal has a substantially higher frequency than the environmental timing signal.

17. A method as claimed in claim **14**, wherein a frequency of the internal clock signal is at least four times higher than a frequency of the environmental timing signal.

18. A method as claimed in claim **14**, further comprising resynthesizing flip-flops in the digital circuit design that are clocked by the environmental timing signal to flip-flops in the new circuit that are clocked by the internal clock signal.

19. A method as claimed in claim **14**, further comprising resynthesizing flip-flops in the digital circuit design that are clocked by the environmental timing signal to flip-flops in the new circuit that are clocked by the internal clock signal and load enabled in response to the environmental timing signal.

20. A method as claimed in claim **14**, further comprising resynthesizing flip-flops in the digital circuit design that are clocked by the environmental timing signal to flip-flops in the new circuit that are clocked by the internal clock signal and load enabled by control signals generated by finite state machines operating in response to the environmental timing signal.

21. A method as claimed in claim **14**, further comprising resynthesizing latches in the digital circuit design that are gated by the environmental timing signal to flip-flops in the new circuit that are clocked by the internal clock signal.

22. A method as claimed in claim 14, further comprising resynthesizing latches in the digital circuit design that are gated by the environmental timing signal to flip-flops in the new circuit that are clocked by the internal clock signal and load enabled in response to the environmental timing signal.

23. A method as claimed in claim 14, further comprising resynthesizing latches in the digital circuit design that are gated by the environmental timing signal to flip-flops in the new circuit that are clocked by the internal clock signal and load enabled by control signals generated by finite state machines operating in response to the environmental timing signal.

24. A method as claimed in claim 14, further comprising performing a simulation of the new circuit.

25. A method as claimed in claim 14, further comprising resynthesizing latches in the digital circuit design that are gated by the environmental timing signal to cascade-connected flip-flops and multiplexers, the multiplexers receiving select signals derived from the environmental timing signal.

26. A method as claimed in claim 25, wherein the select signals received by the multiplexers are generated by a finite state machine controller.

27. A logic system for generating output signals to an environment in response to at least one environmental timing signal and environmental data signals provided from the environment, the logic system comprising:

an internal clock for generating an internal clock signal for the logic system;

logic means for generating the output signals in response to the environmental data signals; and

controller means for coordinating operation of the logic means in response to the internal clock signal and the environmental timing signal.

28. A logic system for generating output signals to an environment in response to at least one environmental timing signal and environmental data signals provided from the environment, the logic system comprising:

an internal clock for generating an internal clock signal for the logic system;

at least one configurable logic device including:

logic which generates the output signals in response to the environmental data signals and the internal clock signal; and

a controller which coordinates operation of the logic in response to the internal clock signal and the environmental timing signal.

29. A logic system as described in claim 28, wherein the controller comprises a synchronizer for sampling the environmental timing signal in response to the internal clock signal.

30. A logic system as described in claim 29, wherein the synchronizer is constructed from non-programmable logic.

31. A logic system as described in claim 29, wherein the controller further comprises a finite state machine for generating control signals to the combinational logic in response to the sampled environmental timing signal.

32. A logic system as described in claim 28, wherein the logic comprises combinational logic and sequential logic.

33. A logic system as described in claim 32, wherein the controller comprises a finite state machine for generating control signals to the sequential logic in response to the environmental timing signal and the internal clock signal.

34. A logic system as described in claim 33, wherein the sequential logic comprises flip-flops receiving the internal clock signal at a clock input and the control signals at a latch enable input.

35. A logic system as described in claim 28, wherein the at least one configurable logic device comprises at least one field programmable gate array.

36. A logic system as described in claim 28, further comprising an interconnect for transmitting signals between plural configurable logic devices.

37. A configurable logic system, comprising:

at least one configurable logic device;

an interconnect providing connections between the logic device and an environment to convey output data signals from the configurable logic device and at least one environmental timing signal and environmental data signals from the environment; and

a configurer for programming the configurable logic device to synchronize the environmental timing signal to an internal clock signal of the logic system.

38. A configurable logic system as described in claim 37, wherein the configurer converts a digital circuit design into a new circuit that is substantially functionally equivalent to the digital circuit design, and programs the at least one configurable logic device with the new circuit.

39. A configurable logic system as described in claim 38, wherein the configurer converts the digital circuit design into the new circuit by resynthesizing sequential logic elements in the digital circuit design that operate in response to the environmental timing signal to operate in response to the internal clock signal in the new circuit.

40. A configurable logic system as described in claim 37, wherein the configurer programs the configurable logic device to have logic and a controller for coordinating operation of the logic in response to the internal clock signal and the environmental timing signal.

41. A configurable logic system as described in claim 40, wherein the configurer programs the controller to include a synchronizer for sampling the environmental timing signal in response to the internal clock signal.

42. A configurable logic system as described in claim 41, wherein the configurer programs the controller to include a finite state machine for generating control signals to the logic in response to the sampled environmental timing signal.

43. A configurable logic system as described in claim 41, wherein the configurer programs the logic to include combinational logic and sequential logic.

44. A configurable logic system as described in claim 41, wherein the configurer programs the controller to include a finite state machine for generating control signals to the sequential logic in response to the environmental timing signal and the internal clock signal.

45. A configurable logic system as described in claim 37, wherein the configurer programs the configurable logic device to include flip-flops that are clocked by the internal clock signal and load enabled in response to the environmental timing signal.

46. A configurable logic system as described in claim 37, wherein the configurer programs the configurable logic device to include:

flip-flops that are clocked by the clock signal and load enabled by control signals; and

finite state machines generating the control signals in response to the environmental timing signal.

47. A configurable logic system as described in claim 37, wherein the environment is a cycle simulation.

48. A configurable logic system as described in claim 37, wherein the environment is a hardware system.

49. A configurable logic system as described in claim 48, wherein the configurable logic system is a logic emulator.

50. A configurable logic system as described in claim 37, wherein the logic system is a simulation accelerator.

* * * * *

U.S. Patent 6,009,531

A427 - 454

US006009531A

United States Patent [19]

Selvidge et al.

[11] Patent Number: 6,009,531

[45] Date of Patent: *Dec. 28, 1999

[54] **TRANSITION ANALYSIS AND CIRCUIT RESYNTHESIS METHOD AND DEVICE FOR DIGITAL CIRCUIT MODELING**

[75] Inventors: **Charles W. Selvidge**, Charlestown; **Matthew L. Dahl**, Cambridge, both of Mass.

[73] Assignee: **Ikos Systems, Inc.**, Cupertino, Calif.

[*] Notice: This patent is subject to a terminal disclaimer.

[21] Appl. No.: **08/863,963**

[22] Filed: **May 27, 1997**

Related U.S. Application Data

[63] Continuation of application No. 08/513,605, Aug. 10, 1995, Pat. No. 5,649,176.

[51] **Int. Cl.**[6] **G06F 1/12**

[52] **U.S. Cl.** **713/400**

[58] **Field of Search** 713/400; 710/8, 710/10, 12; 712/10, 15, 36, 37

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,450,560	5/1984	Conner	371/25
4,697,241	9/1987	Lavi	364/488
5,513,338	4/1996	Alexander et al.	395/500
5,572,710	11/1996	Asano et al.	395/500

FOREIGN PATENT DOCUMENTS

0 453 171 A2	10/1991	European Pat. Off.	G06F 1/04
2 180 382	3/1987	United Kingdom	H03K 19/00

OTHER PUBLICATIONS

Laird, D., et al., "Delay Compensator," *LSI Logic Corp.*, pp. 1–8, (Aug. 1990).

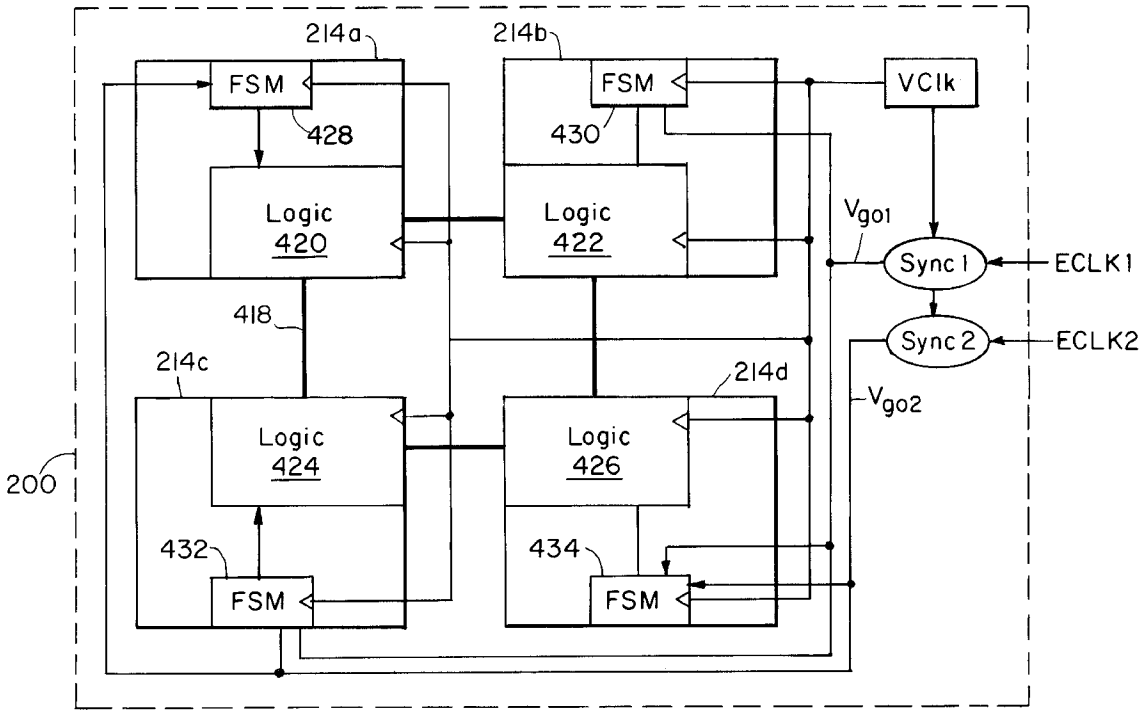
Primary Examiner—Thomas M. Heckler

Attorney, Agent, or Firm—Hamilton, Brook, Smith & Reynolds, P.C.

[57] **ABSTRACT**

A method of configuring a configurable logic system, including a single or multi-FPGA network, is disclosed in which an internal clock signal is defined that has a higher frequency than timing signals the system receives from the environment in which it is operating. The frequency can be at least ten times higher than a frequency of the environmental timing signals. The logic system is configured to have a controller that coordinates operation of its logic operation in response to the internal clock signal and environmental timing signals. Specifically, the controller is a finite state machine that provides control signals to sequential logic elements such as flip-flops. The logic elements are clocked by the internal clock signal. In the past, emulation or simulation devices, for example, operated in response to timing signals from the environment. A new internal clock signal, invisible to the environment, rather than the timing signals is used to control the internal operations of the devices. Additionally, a specific set of transformations are disclosed that enable the conversion of a digital circuit design with an arbitrary clocking methodology into a single clock synchronous circuit.

19 Claims, 14 Drawing Sheets



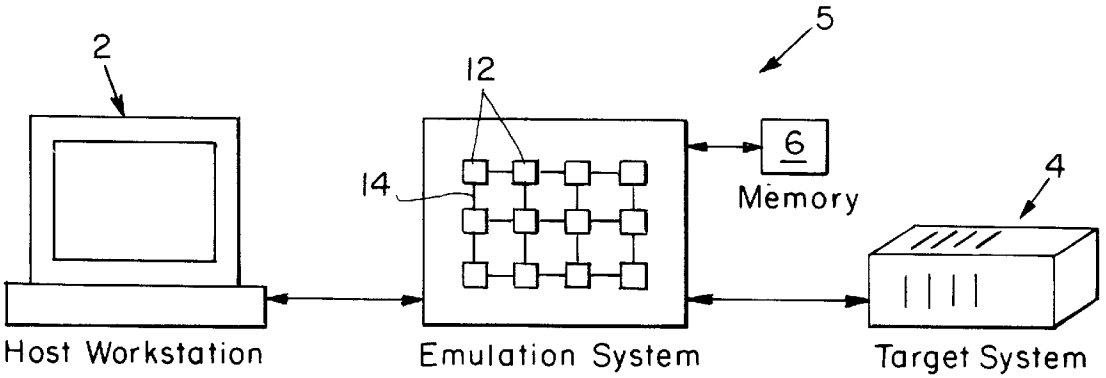


FIG. 1
(Prior Art)

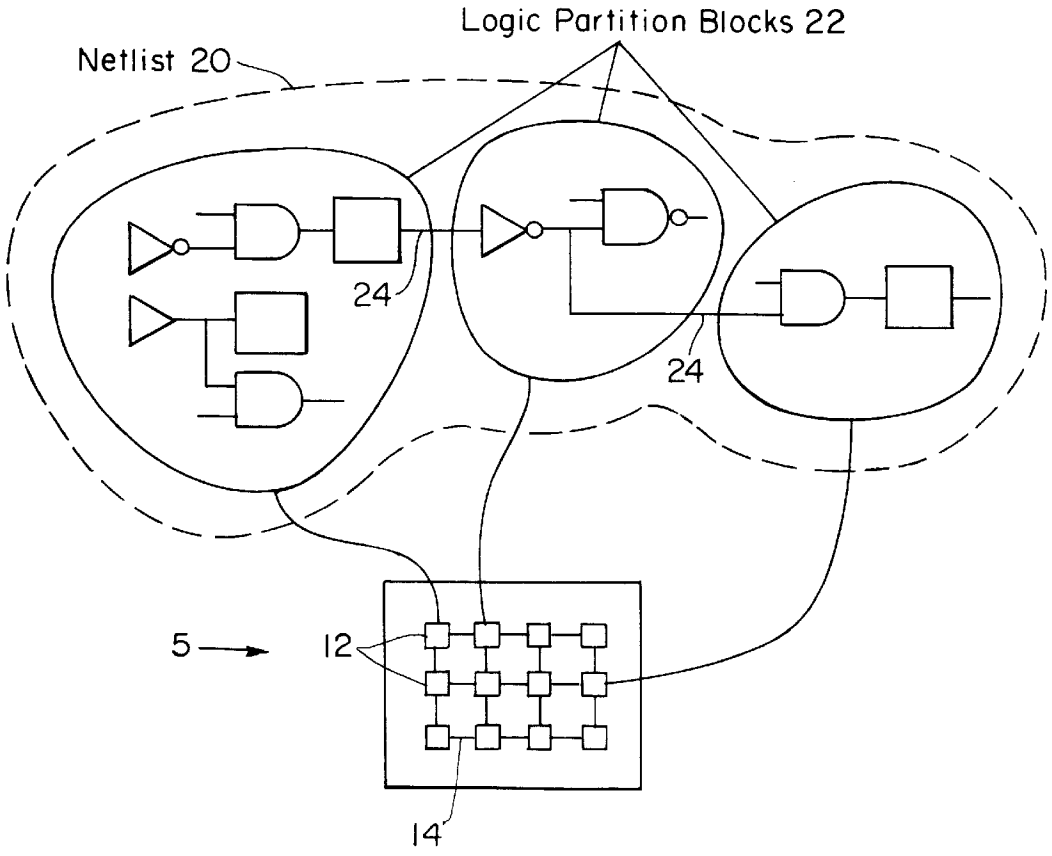


FIG. 2

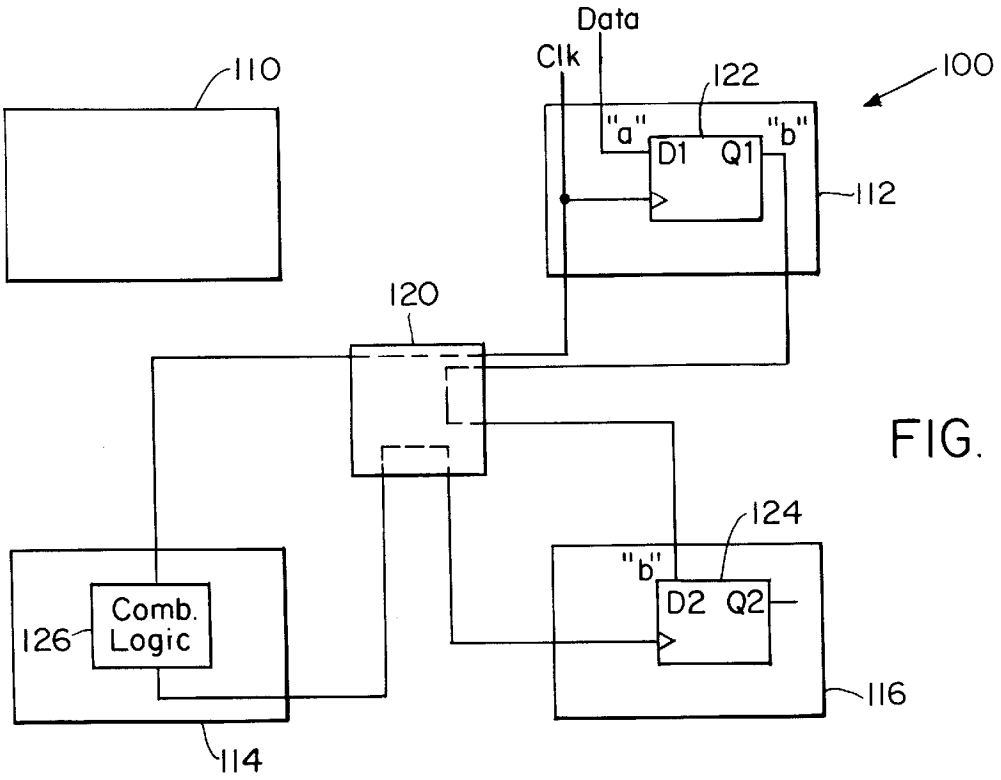

Data
Clk
110
122
100
"a"
"b"
D1
Q1
112
120
124
"b"
D2
Q2
Comb.
Logic
126
116
114

FIG. 3

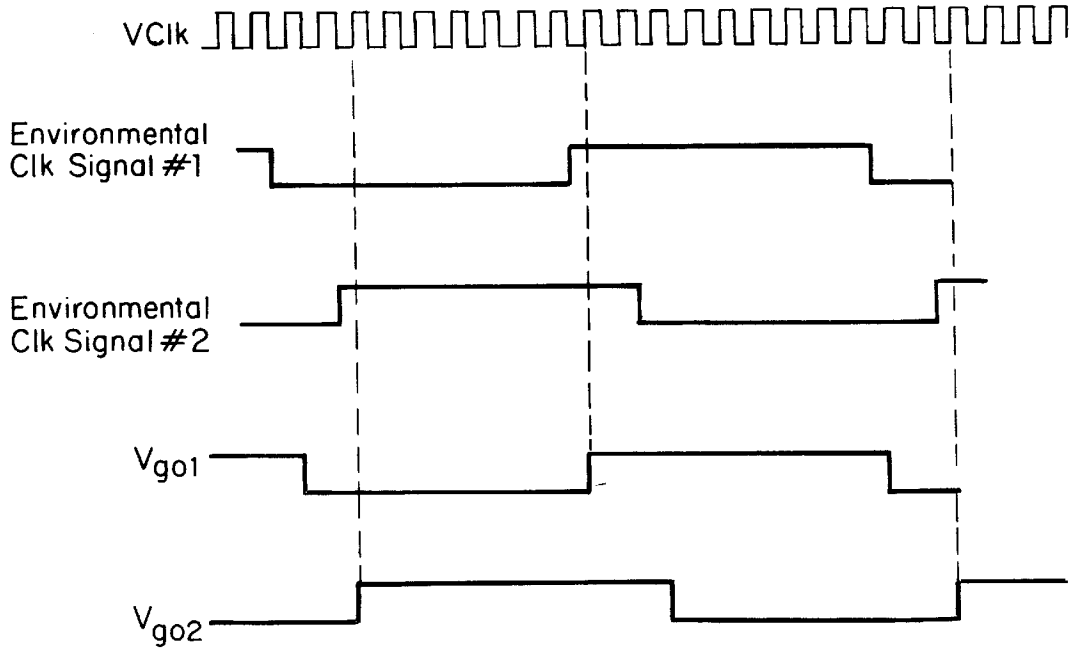

VClk
Environmental
Clk Signal #1
Environmental
Clk Signal #2
Vgo1
Vgo2

FIG. 4B

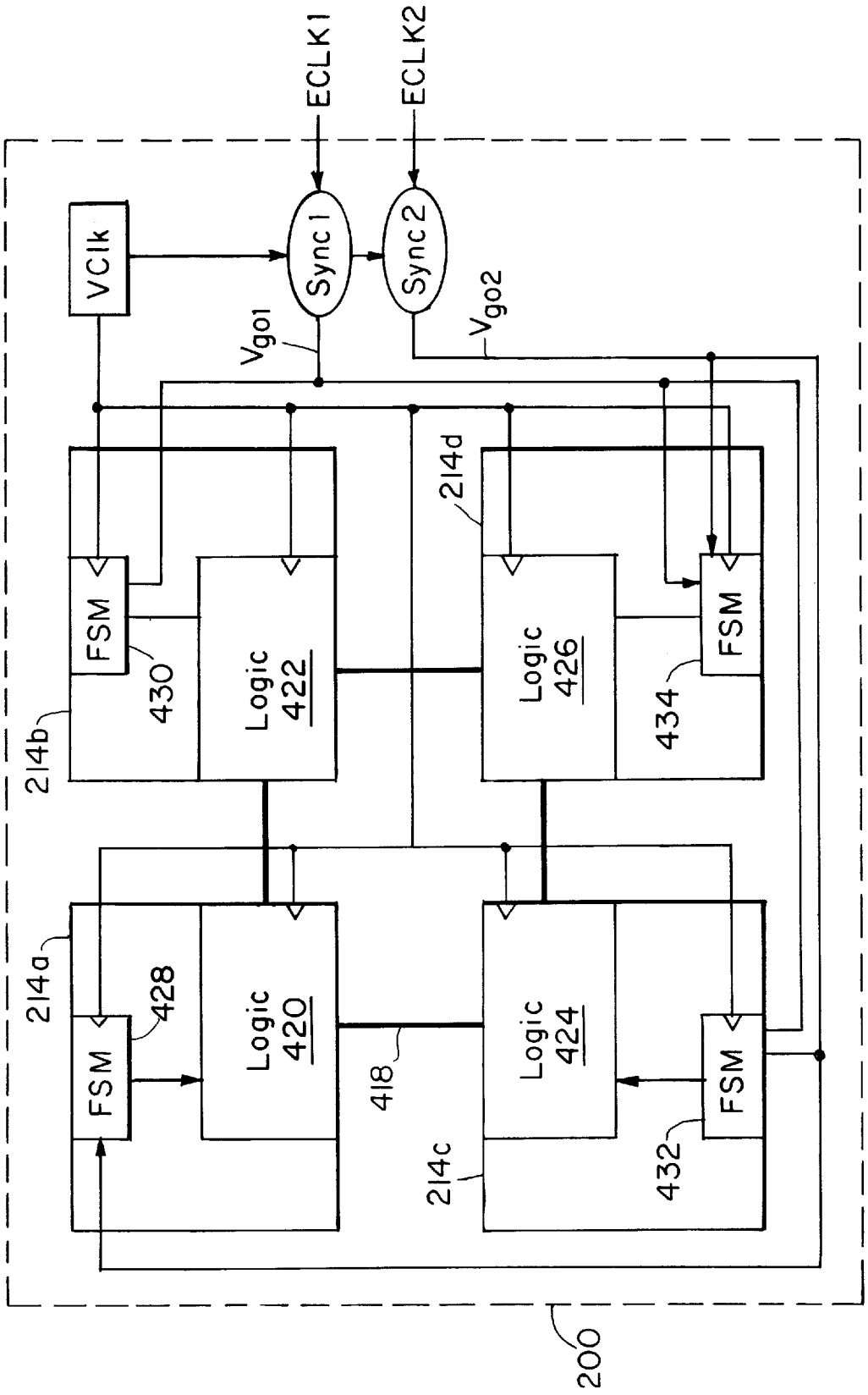

ECLK1
ECLK2
VClk
Sync 1
Sync 2
V_{go1}
V_{go2}
214d
FSM
430
Logic 422
214b
Logic 426
434
FSM
214a
428
FSM
Logic 420
418
Logic 424
214c
432
FSM
200

FIG. 4A

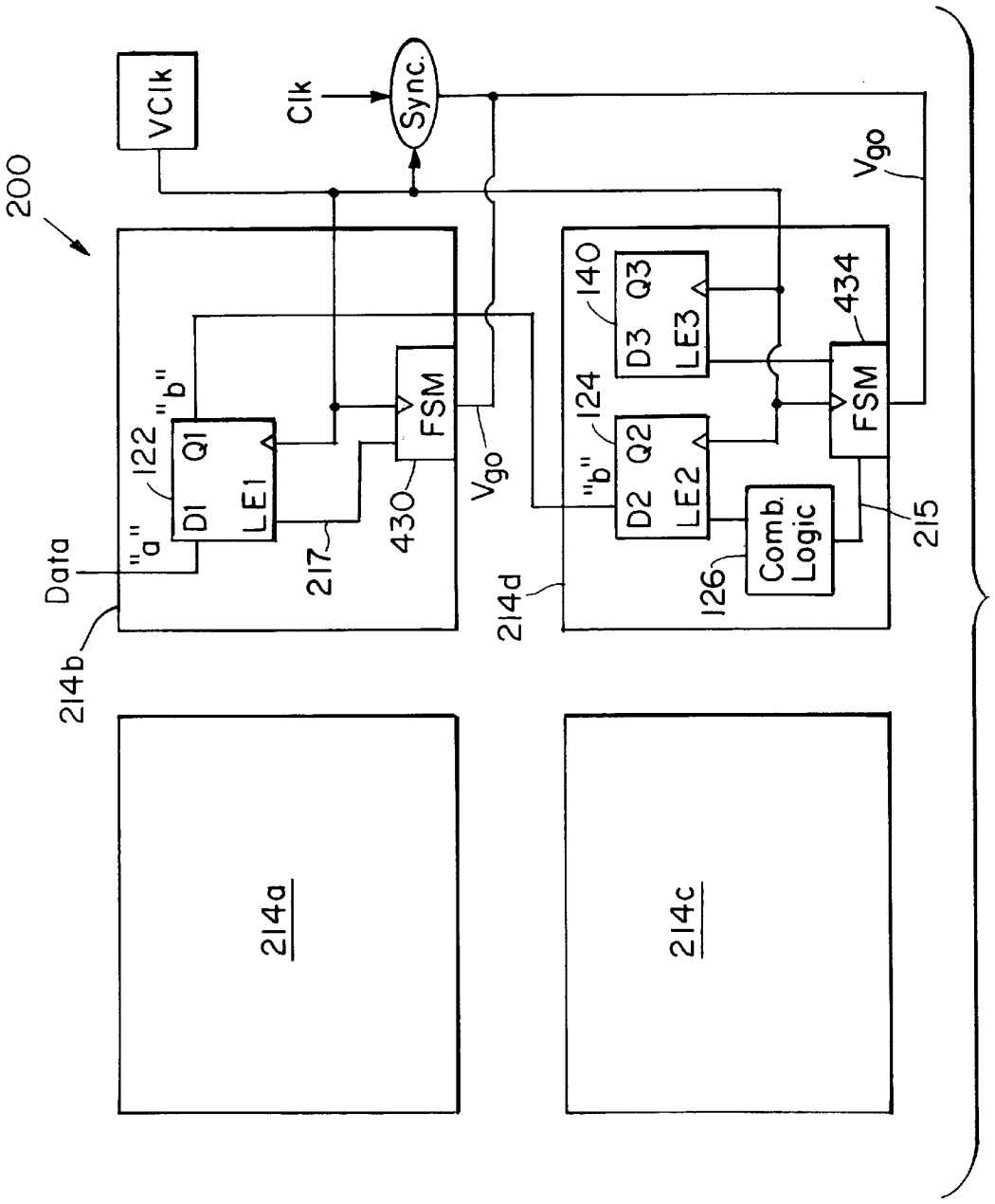

200
VClk
Clk
Sync.
Vgo
140
D3 Q3
LE3
434
FSM
124
"b"
D2 Q2
LE2
Comb. Logic
215
126
214d
214c
122
"b"
D1 Q1
LE1
"a"
Data
214b
217
430
FSM
Vgo
214a

FIG. 5A

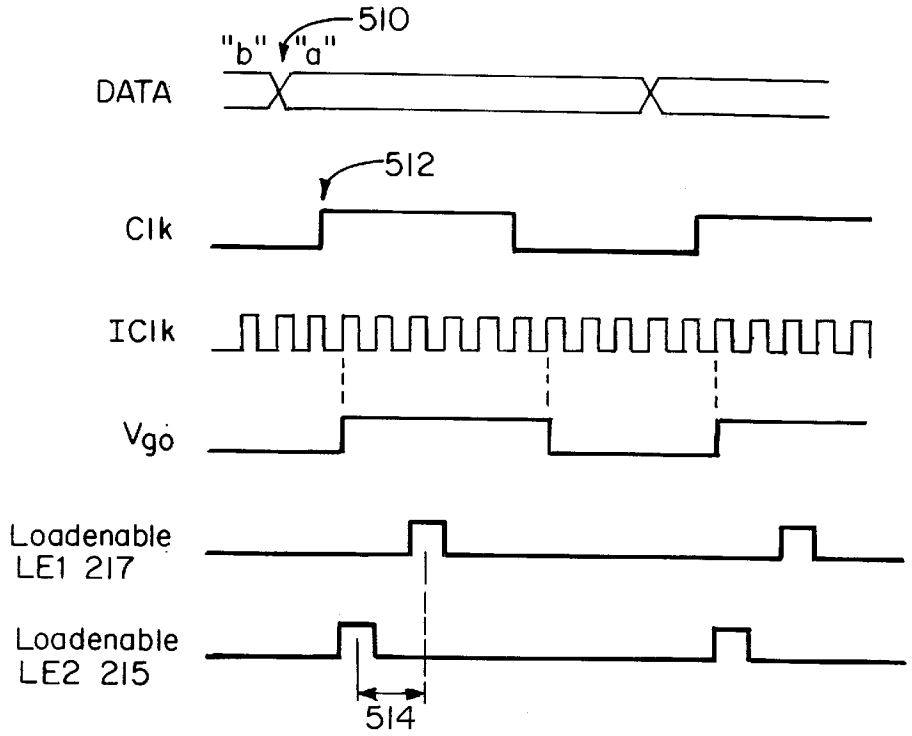

510
"b"
"a"
DATA
512
Clk
IClk
Vgo
Loadenable
LE1 217
Loadenable
LE2 215
514

FIG. 5B

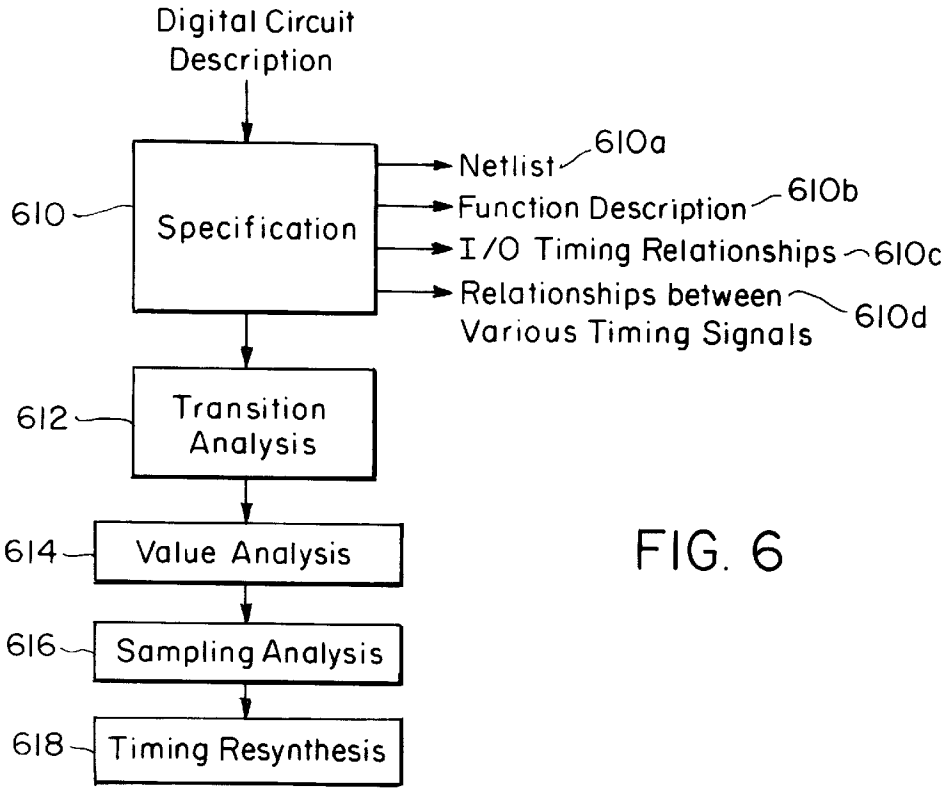

Digital Circuit
Description
610
Specification
Netlist
610a
Function Description
610b
I/O Timing Relationships
610c
Relationships between
Various Timing Signals
610d
612
Transition
Analysis
614
Value Analysis
616
Sampling Analysis
618
Timing Resynthesis

FIG. 6

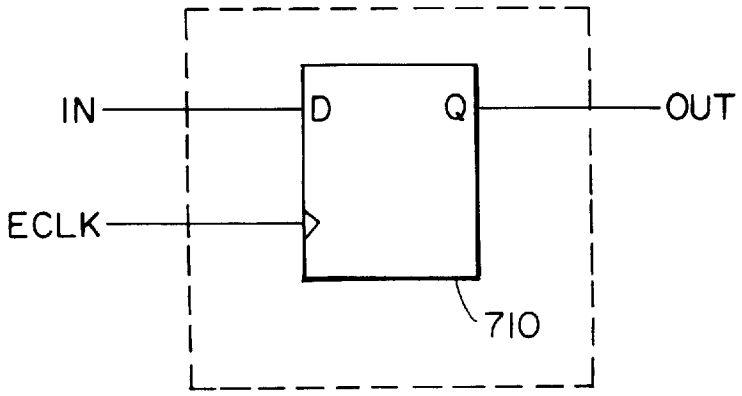

IN
D
Q
OUT
ECLK
710

FIG. 7A

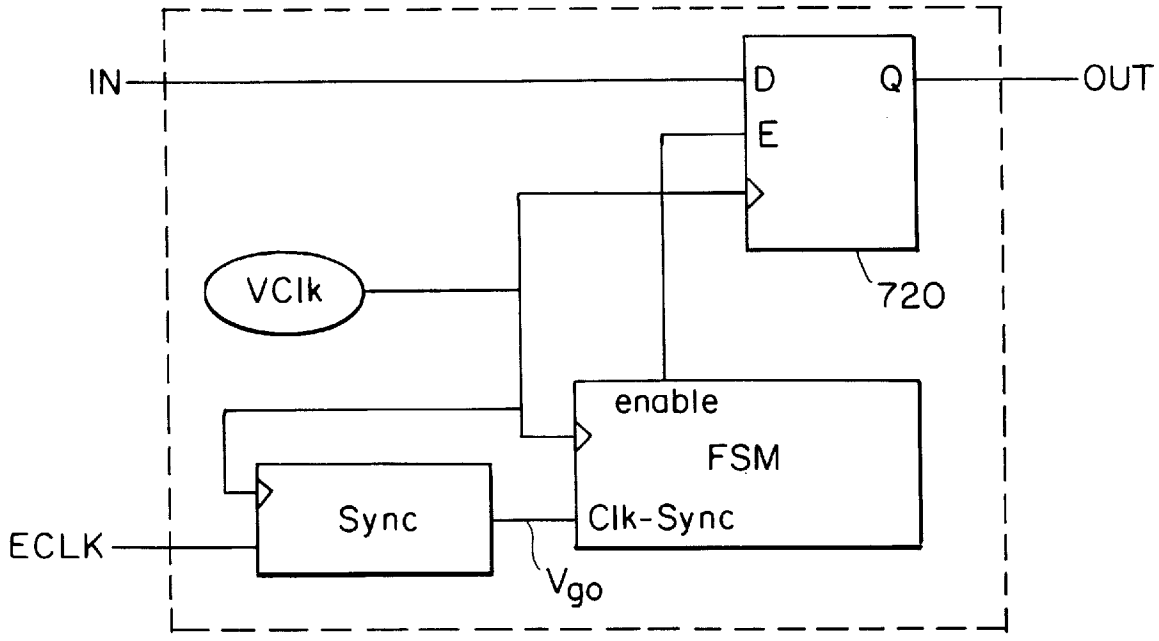

IN
D
Q
OUT
E
VClk
720
enable
FSM
Sync
Clk-Sync
ECLK
Vgo

FIG. 7B

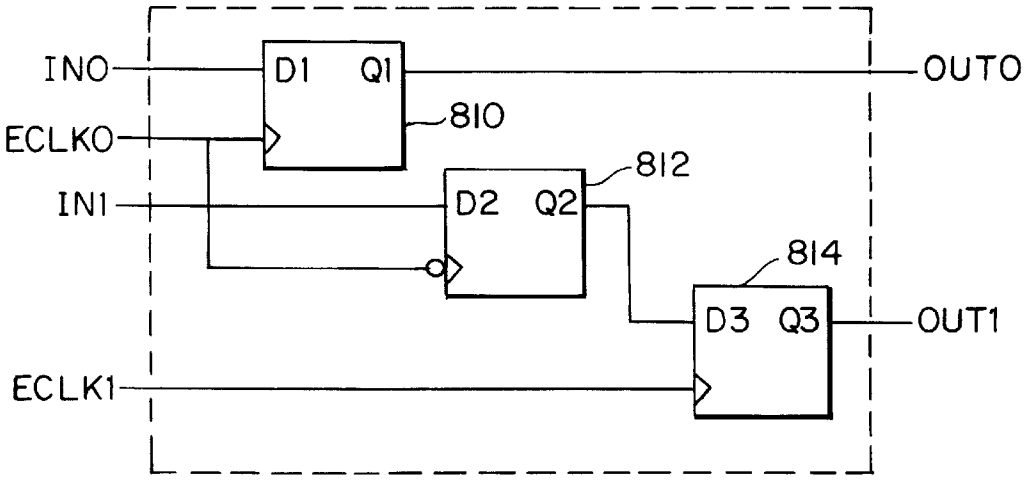

INO
ECLKO
IN1
ECLK1
D1 Q1
810
D2 Q2
812
D3 Q3
814
OUTO
OUT1

FIG. 8A

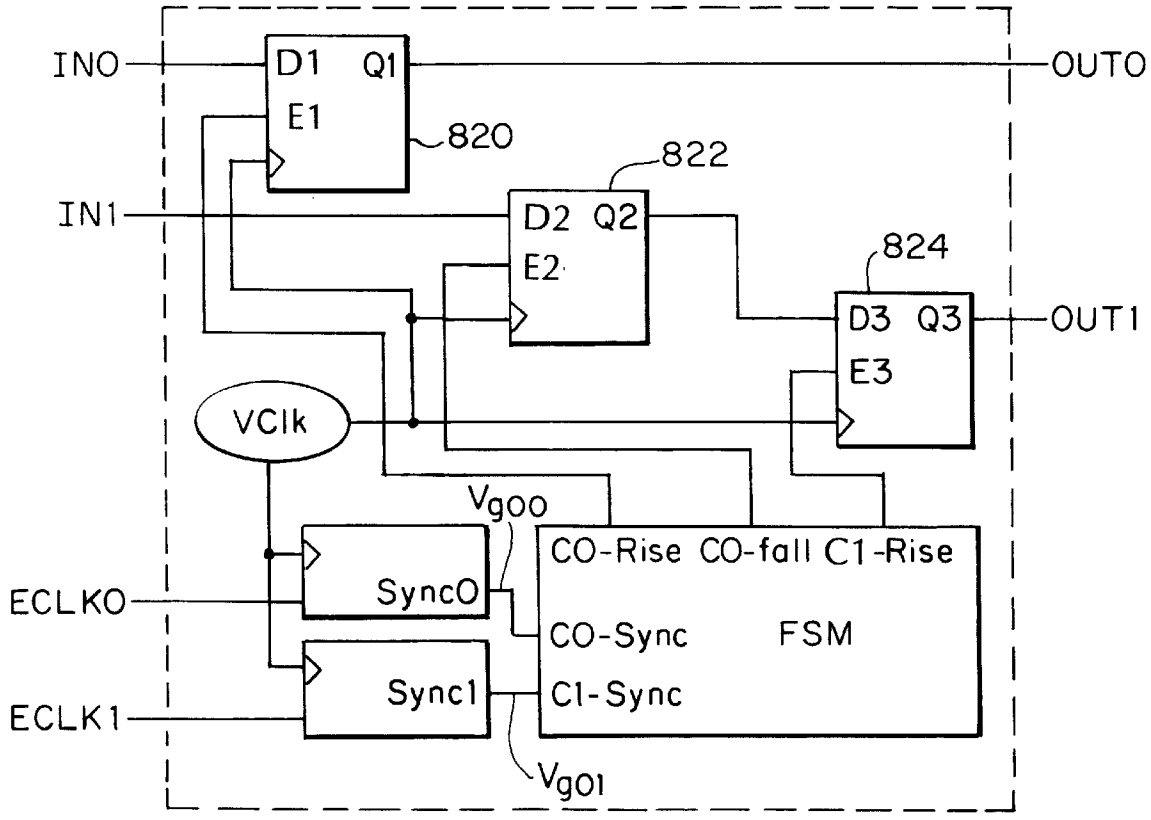

INO
IN1
ECLKO
ECLK1
D1 Q1
E1
820
D2 Q2
E2
822
D3 Q3
E3
824
OUTO
OUT1
VClk
Vgoo
Vgo1
SyncO
Sync1
CO-Rise CO-fall C1-Rise
CO-Sync
C1-Sync
FSM

FIG. 8B

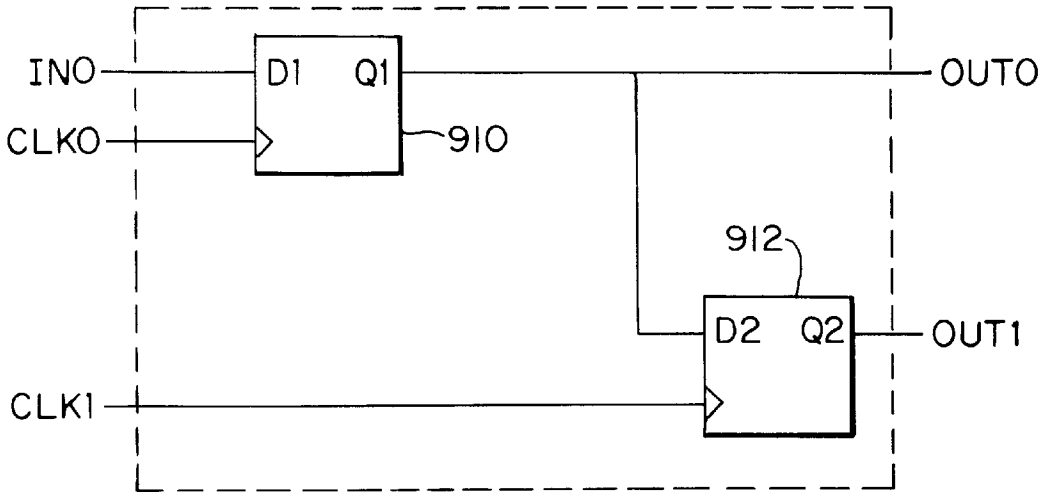

INO
CLKO
D1
Q1
910
OUTO
912
D2
Q2
OUT1
CLK1

FIG. 9A

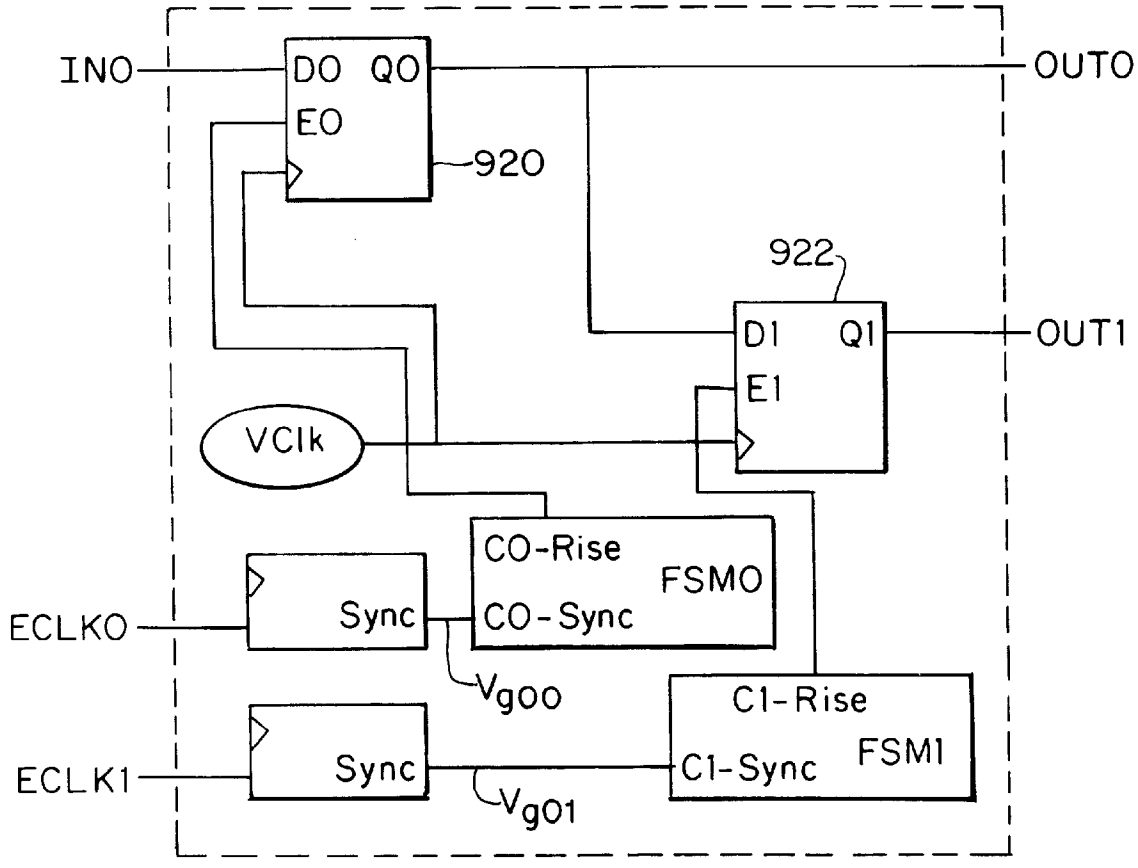

INO
DO
QO
EO
920
OUTO
922
D1
Q1
E1
OUT1
VClk
CO-Rise
FSMO
CO-Sync
Sync
ECLKO
Vgoo
C1-Rise
FSM1
C1-Sync
Sync
ECLK1
Vgo1

FIG. 9B

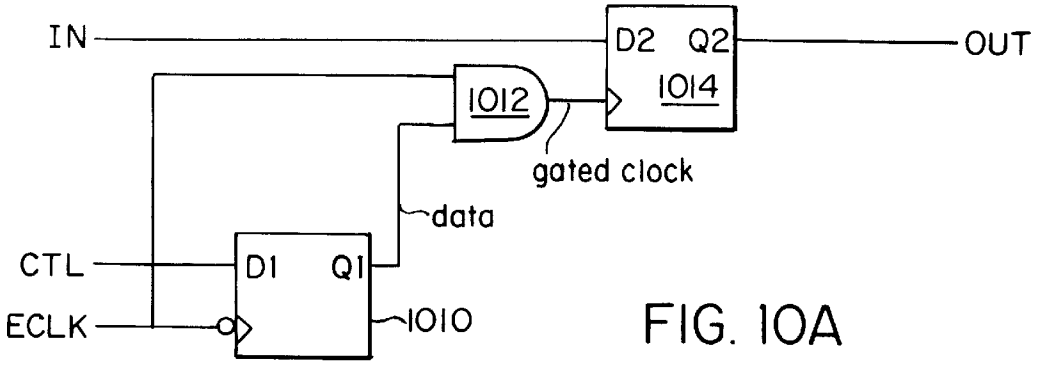


FIG. 10A

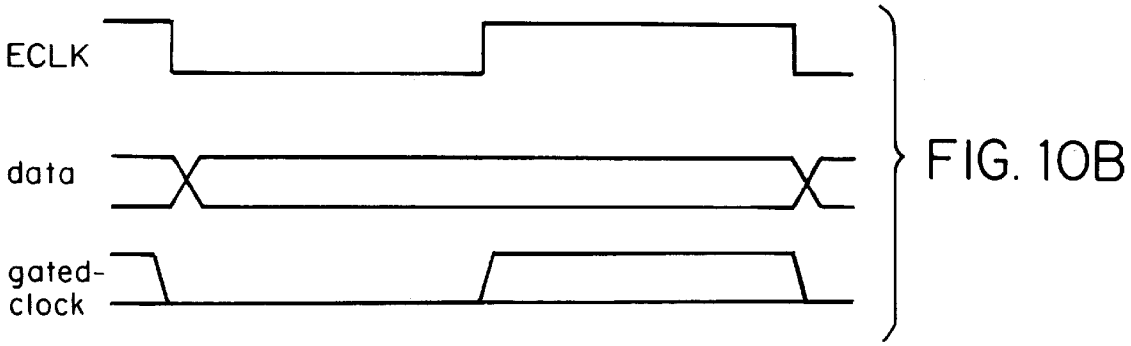


FIG. 10B

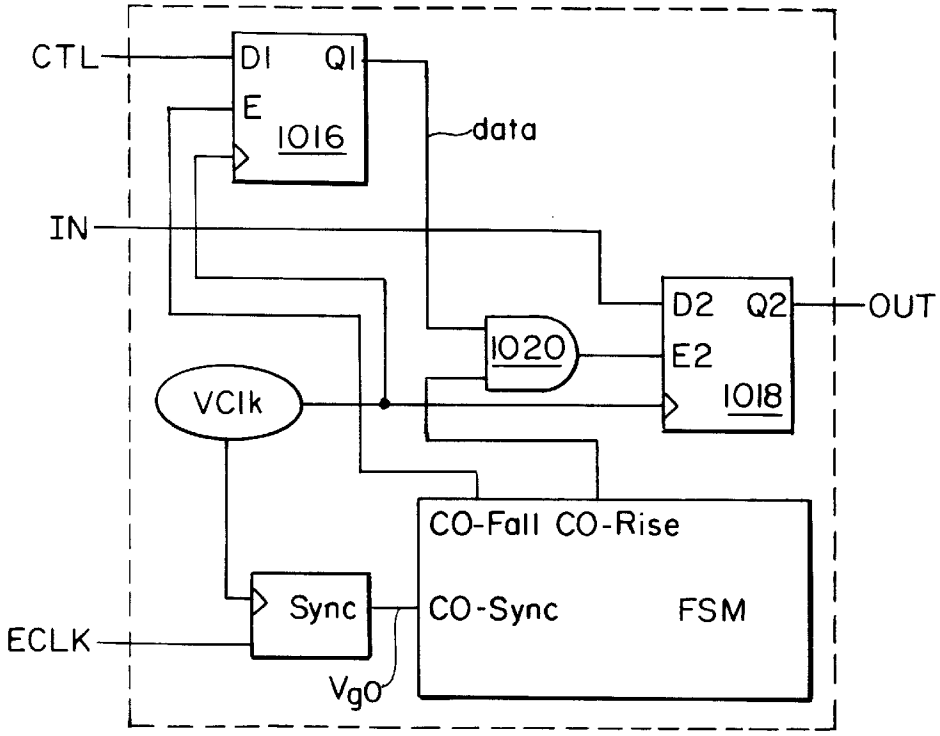


FIG. 10C

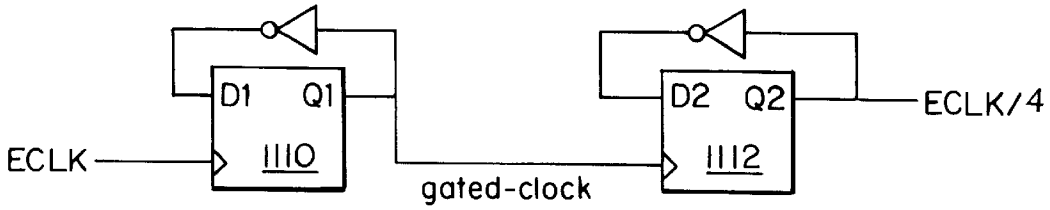


FIG. 11A

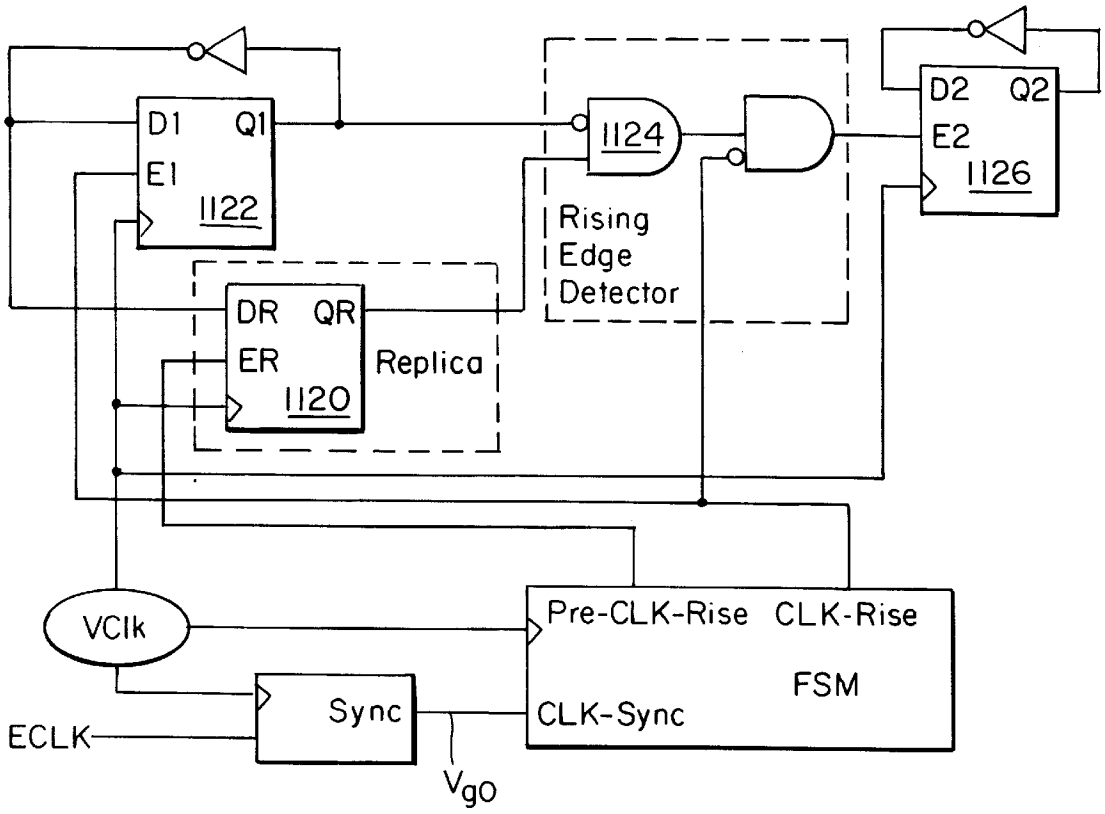


FIG. 11B

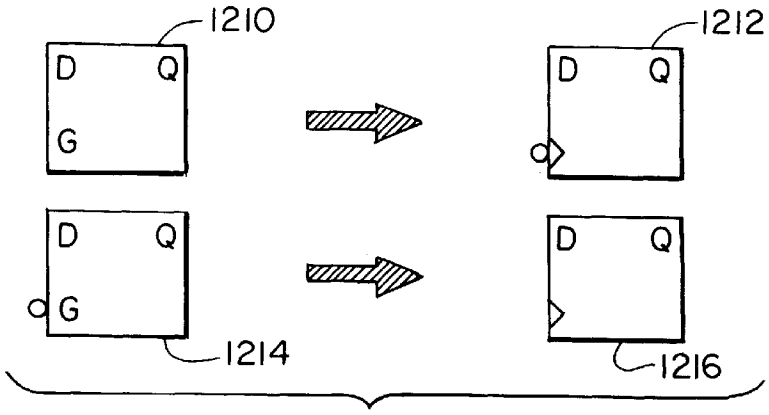


FIG. 12

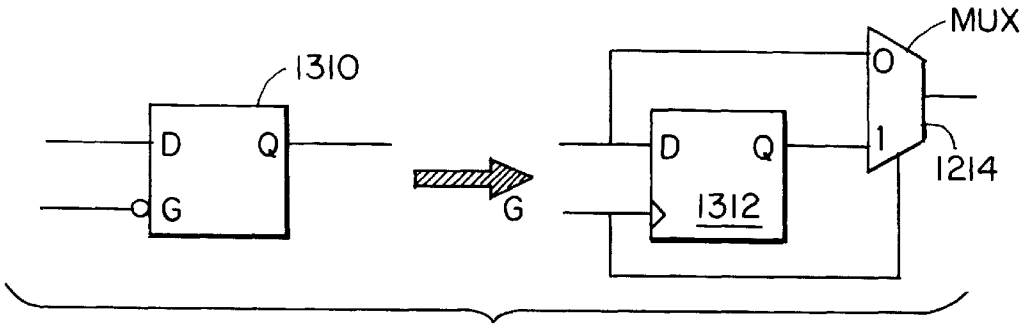


FIG. 13

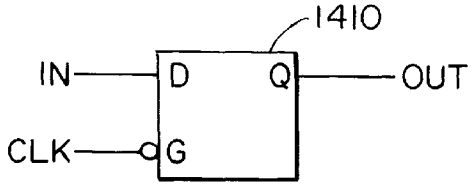


FIG. 14A

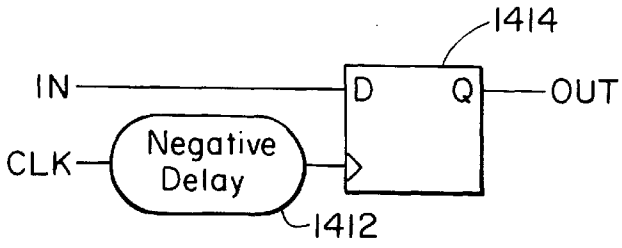


FIG. 14B

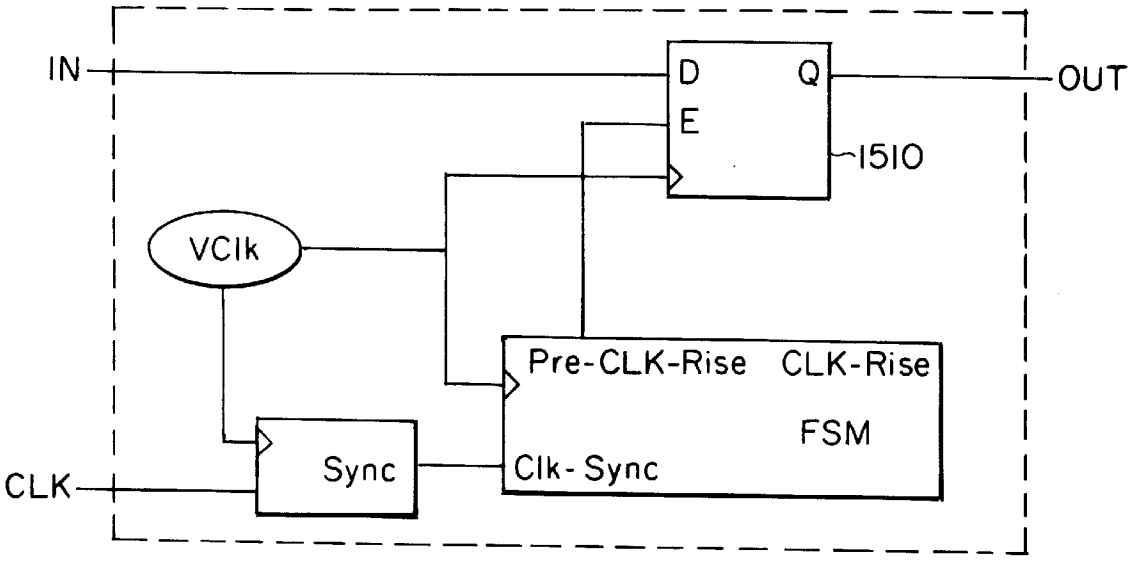


FIG. 15

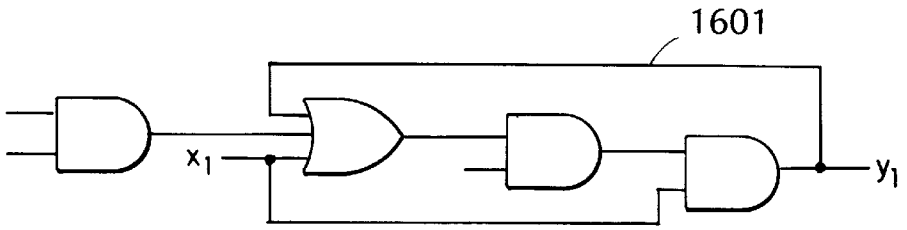


FIG. 16A

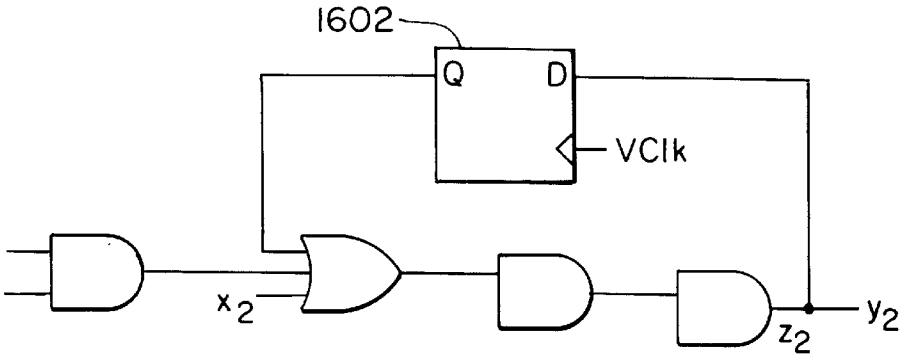


FIG. 16B

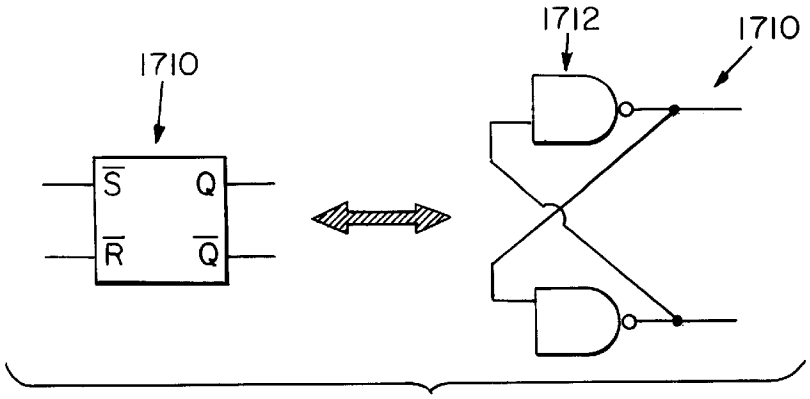


FIG. 17A

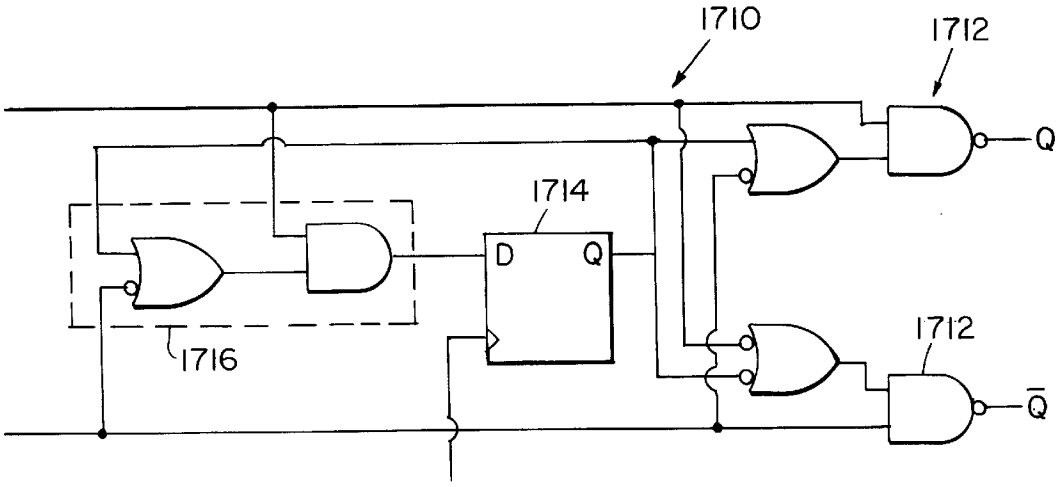


FIG. 17B

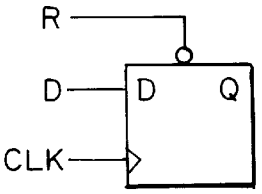


FIG. 18A

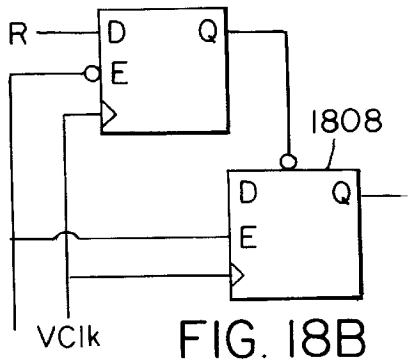


FIG. 18B

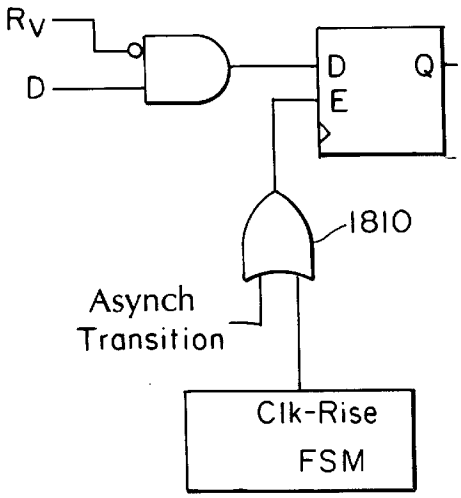


FIG. 18C

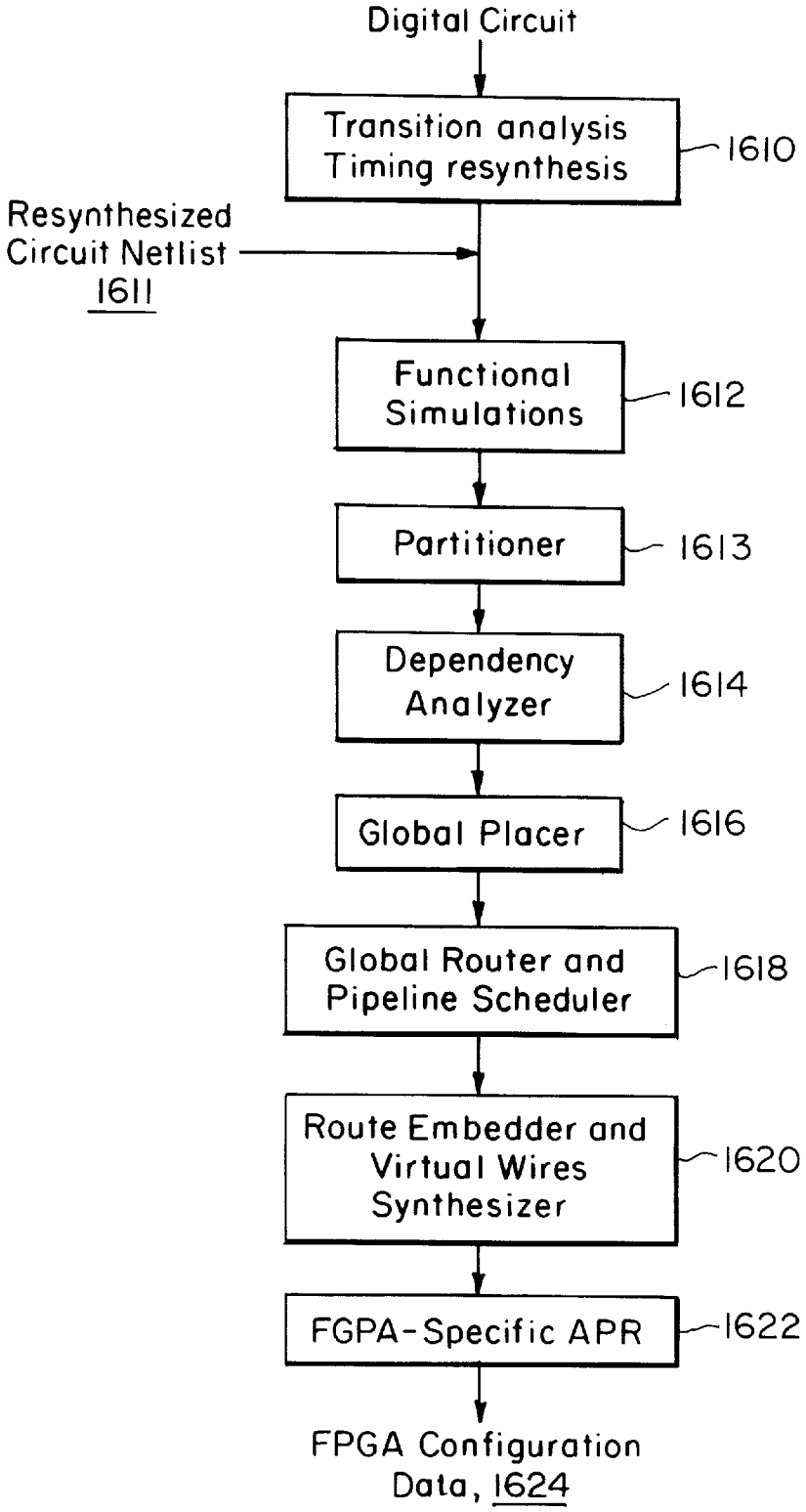

Digital Circuit
Transition analysis
Timing resynthesis
1610
Resynthesized
Circuit Netlist
1611
Functional
Simulations
1612
Partitioner
1613
Dependency
Analyzer
1614
Global Placer
1616
Global Router and
Pipeline Scheduler
1618
Route Embedder and
Virtual Wires
Synthesizer
1620
FGPA-Specific APR
1622
FPGA Configuration
Data, 1624

FIG. 19

TRANSITION ANALYSIS AND CIRCUIT RESYNTHESIS METHOD AND DEVICE FOR DIGITAL CIRCUIT MODELING

RELATED APPLICATION

This application is a continuation of application Ser. No. 08/513,605 filed Aug. 10, 1995, now U.S. Pat. No. 5,649,176, which is incorporated herein by reference in its entirety.

BACKGROUND OF THE INVENTION

Configurable logic devices are a general class of electronic devices that can be easily configured to perform a desired logic operation or calculation. One example is Mask Programmed Gate Arrays (MPGA). These devices offer density and performance. Poor turn around time coupled with only one-time configurability tend to diminish their ubiquitous use. Reconfigurable logic devices or programmable logic devices (such as Field Programmable Gate Arrays (FPGA)) offer lower levels of integration but are reconfigurable, i.e., the same device may be programmed many times to perform different logic operations. Most importantly, the devices can be programmed to create gate array prototypes instantaneously, allowing complete dynamic reconfigurability, something that MPGAs can not provide.

System designers commonly use reconfigurable logic devices such as FPGAs to test logic designs prior to manufacture or fabrication in an effort to expose design flaws. Usually, these tests take the form of emulations in which a reconfigurable logic devices models the logic design, such as a microprocessor, in order to confirm the proper operation of the logic design along with possibly its compatibility with an environment or system in which it is intended to operate.

In the case of testing a proposed microprocessor logic design, a netlist describing the internal architecture of the microprocessor is compiled and then loaded into a particular reconfigurable logic device by some type of configuring device such as a host workstation. If the reconfigurable logic device is a single or array of FPGAs, the loading step is as easy as down-loading a file describing the compiled netlist to the FPGAs using the host workstation or other computer. The programmed configurable logic device is then tested in the environment of a motherboard by confirming that its response to inputs agrees with the design criteria.

Alternatively, reconfigurable logic devices also find application as hardware accelerators for simulators. Rather than testing a logic design by programming a reconfigurable device to "behave" as the logic device in the intended environment for the logic design, e.g., the motherboard, a simulation involves modeling the logic design on a workstation. In this environment, the reconfigurable logic device performs gate evaluations for portions of the model in order to relieve the workstation of this task and thereby decreases the time required for the simulation.

Recently, most of the attention in complex logic design modeling has been directed toward FPGAs. The lower integration of the FPGAs has been overcome by forming heterogeneous networks of special purpose FPGA processors connected to exchange signals via some type of interconnect. The network of the FPGAs is heterogeneous not necessarily in the sense that it is composed of an array of different devices but that the devices have been individually configured to cooperatively execute different sections, or partitions, of the overall logic design. These networks rely on static routing at compile-time to organize the propagation of logic signals through the FPGA network. Static refers to the fact that all data or logic signal movement can be determined and optimized during compiling.

When a logic design intended for eventual MPGA fabrication is mapped to an FPGA, hold time errors are a problem that can arise, particularly in these complex configurable logic device networks. A digital logic design that has been loaded into the configurable logic devices receives timing signals, such as clock signals, and data signals from the environment in which it operates. Typically, these timing signals coordinate the operation of storage or sequential logic components such as flip-flops or latches. These storage devices control the propagation of combinational signals, which are originally derived from the environmental data signals, through the logic devices.

Hold time problems commonly arise where a timing signal is intended to clock a particular storage element to signal that a value at the element's input terminal should be held or stored. As long as the timing signal arrives at the storage element while the value is valid, correct operation is preserved. Hold time violations occur when the timing signal is delayed beyond a time for which the value is valid, leading to the loss of the value. This effect results in the destruction of information and generally leads to the improper operation of the logic design.

Identification and mitigation of hold time problems presents many challenges. First, while the presence of a hold time problem can be recognized by the improper operation of the logic design, identifying the specific location within the logic design of the hold time problem is a challenge. This requires sophisticated approximations of the propagation delays of timing signals and combinational signals through the logic design. Once a likely location of a hold time problem has been identified, he typical approach is somewhat ad hoc. Delay is added on he path of the combinational signals to match the timing signal delays. This added delay, however, comes at its own cost. First, the operational speed of the design must now take into account this new delay. Also, new hold time problems can now arise because of the changed clock speed. In short, hold time problems are both difficult to identify and then difficult to rectify.

Other problems arise when a logic design intended for ultimate MPGA fabrication, for example, is realized in FPGAs. Latches, for instance, are often implemented in MPGAs. FPGA, however, do not offer a corresponding element.

SUMMARY OF THE INVENTION

The present invention seeks to overcome the hold time problem by imposing a new timing discipline on a given digital circuit design through a resynthesis process that yields a new but equivalent circuit. The resynthesis process also transforms logic devices and timing structures to those that are better suited to FPGA implementation. This new timing discipline is insensitive to unpredictable delays in the logic devices and eliminates hold time problems. It also allows efficient implementation of latches, multiple clocks, and gated clocks. By means of the resynthesis, the equivalent circuit relies on a new higher frequency internal clock (or virtual clock) that is distributed with minimal skew. The internal clock signal controls the clocking of all or substantially all the storage elements, e.g. flip-flops, in the equivalent circuit, in effect discretizing time and space into manageable pieces. The user's clocks are treated in the same manner as user data signals.

In contrast with conventional approaches, the present invention does not allow continuous inter-FPGA signal flow.

Instead, all signal flow is synchronized to the internal clock so that signals flow between flip-flops through intermediate FPGAs in discrete hops. The internal clock provides a time base for the circuit's operation.

In general, according to one aspect, the invention features a method of configuring a configurable or programmable logic system. Generally, such logic systems include single or multi-FPGA network, although the invention can be applied to other types of configurable devices. Particular to the invention, the logic system is provided with an internal clock signal that typically has a higher frequency, by a factor of at least four, than timing signals the system receives from the environment in which it is operating. The logic system is configured to have a controller that coordinates operation of the logic in response to the internal clock signal and the environmental timing signals. In the past, while emulation or simulation devices, for example, operated in response to timing signals from the environment, a new internal clock signal, invisible to the environment, was not used to control the internal operations of the devices.

In specific embodiments, a synchronizer is incorporated to essentially generate a synchronized version of the environmental timing signal. This synchronized version behaves much like other data signals from the environment. This synchronizer feeds the resulting sampled environmental clock signals to a finite state machine, which generates control signals. The logic operations are then coordinated by application of these control signals to sequential logic elements.

In more detail, the logic system is configured to have both combinational logic, e.g. logic gates, and sequential logic, e.g. flip-flops, to perform the logic operations. The control signals function as load enable signals to the sequential logic. The internal clock signal is received at the clock terminals of that logic. Just like the original digital circuit design, each sequential logic element operates in response to the environmental timing signals. Now, however, these timing signal control the load enable of the elements, not the clocking. It is the internal clock signal that now clocks the elements. As a result, the resynthesized circuit operates synchronously with a single clock signal regardless of the clocking scheme of the original digital circuit.

In general, according to another aspect, the invention features a method for converting a digital circuit design into a new circuit that is substantially functionally equivalent to the digital circuit design. First, the internal clock signal is defined, then sequential logic elements of the digital circuit design are resynthesized to operate in response to the internal clock signal in the new circuit rather than simply the environmental timing signals.

In specific embodiments, flip-flops of the digital circuit design, which are clocked by the environmental timing signal, are resynthesized to be clocked by the internal clock signal and load enabled in response to the environmental timing signals. Finite state machines are used to actually generate control signals that load enable each flip-flop. The load enable signals are sometimes also generated from a logic combination of finite state machine signals and logic gates.

In other embodiments, latches in the digital circuit design, which were gated by the environmental timing signals, are resynthesized to be flip-flops or latches in future FPGA designs in the new circuit that are clocked by the new internal or virtual clock signal. These new flip-flops are load enabled in response to the environmental timing signals.

In general, according to still another aspect, the invention features a logic system for generating output signals to an

environment in response to at least one environmental timing signal and environmental data signals provided from the environment. This logic system has its own internal clock and at least one configurable logic device. The internal architecture of the configurable device includes logic for generating the output signals in response to the environmental data signals and a controller, specifically a finite state machine, for coordinating operation of the logic in response to the internal clock signal and the environmental timing signal.

Specifically, the logic includes sequential and combinational logic elements. The sequential logic elements are clocked by the internal clock signal and load enabled in response to the environmental timing signals.

The above and other features of the invention including various novel details of construction and combinations of parts, and other advantages, will now be more particularly described with reference to the accompanying drawings and pointed out in the claims. It will be understood that the particular method and device embodying the invention is shown by way of illustration and not as a limitation of the invention. The principles and features of this invention may be employed in various and numerous embodiments without the departing from the scope of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

In the accompanying drawings, like reference characters refer to the same parts throughout the different views. The drawings are not necessarily to scale and in some cases have been simplified. Emphasis is instead placed upon illustrating the principles of the invention. Of the drawings:

FIG. **1** is a schematic diagram showing a prior art emulation system and its interaction with an environment and a host workstation;

FIG. **2** shows a method for impressing a logic design on the emulation system;

FIG. **3** is a schematic diagram of a configurable logic system that comprises four configurable logic devices—a portion of the internal logic structure of these devices has been shown to illustrate the origins of hold time violations;

FIG. **4A** is a schematic diagram of the logic system of the present invention showing the internal organization of the configurable logic devices and the global control of the logic devices by the internal or virtual clock;

FIG. **4B** is a timing diagram showing the timing relationships between the internal or virtual clock signal, environmental timing signals, and control signals generated by the logic system;

FIG. **5A** is a schematic diagram of a logic system of the present invention that comprises four configurable logic devices, the internal structure of these devices is the functional equivalent of the structure shown in FIG. **3** except that the circuit has been resynthesized according to the principles of the present invention;

FIG. **5B** is a diagram showing the timing relationship between the signals generated in the device of FIG. **5A**;

FIG. **6** illustrates a method by which a digital circuit description having an arbitrary clocking methodology is resynthesized into a functionally equivalent circuit that is synchronous with a single internal clock;

FIGS. **7A** and **7B** illustrate a timing resynthesis circuit transformation in which an edge-triggered flip-flop is converted into a load-enable type flip-flop;

FIGS. **8A** and **8B** illustrate a timing resynthesis circuit transformation in which a plurality of edge triggered flip-

flops clocked by two phase-locked clock signals are converted into load enable flip-flops that are synchronous with the internal clock signal;

FIGS. **9**A and **9**B illustrate a timing resynthesis circuit transformation in which two edge triggered flip-flops clocked by two arbitrary clock signals are transformed into load enabled flip-flops that operate synchronously with the internal clock signal;

FIGS. **10**A, **10**B, and **10**C illustrate a timing resynthesis circuit transformation in which two edge-triggered flip-flops, one of which is clocked by a gated clock, are transformed into two load-enable flip-flops that operate synchronously with the internal clock signal, FIG. **10**B is a timing diagram showing the signal values over time in the circuit;

FIGS. **11**A and **11**B illustrate a timing resynthesis circuit transformation in which a complex gated clock structure, with a second flip-flop being clocked by a gated clock, is converted into a circuit containing three flip-flops and an edge detector, all of the flip-flops operating off of the internal clock signal in the new circuit;

FIG. **12** illustrates circuit transformations in which gated latches are converted into edge-triggered flip-flops on the assumption that the latches are never sampled when open, i.e., latch output is not registered into another storage element when they are open;

FIG. **13** illustrates a timing resynthesis circuit transformation in which a gated latch is converted into an edge-triggered flip-flop and a multiplexor;

FIGS. **14**A and **14**B illustrate a timing resynthesis circuit transformation in which a latch is converted to an edge-triggered flip-flop with a negative delay at the clock input terminal to avoid glitches;

FIG. **15** illustrates a timing resynthesis circuit transformation of the negative delay flip-flop of FIG. **14**B into a flip-flop that operates synchronously with the internal clock signal;

FIGS. **16**A and **16**B illustrate a timing resythesis circuit transformation in which a flip-flop is inserted in a combinational loop to render the circuit synchronous with the virtual clock;

FIGS. **17**A and **17**B illustrate a timing resynthesis circuit transformation in which an RS flip-flop is transformed into a device that is synchronous with the virtual clock;

FIGS. **18**A, **18**B, and **18**C illustrate a timing resynthesis circuit transformation for handling asynchronous preset and clears of state elements; and

FIG. **19** illustrates the steps performed by a compiler that resynthesizes the digital circuit design and converts it into FPGA configuration data that is loaded into the logic system **200**.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Turning now to the drawings, FIG. **1** is a schematic diagram showing an emulation system **5** of the prior art. The emulation system **5** operates in an environment such as a target system **4** from which it receives environmental timing signals and environmental data signals and responsive to these signals generates output data signals to the environment. A configuring device **2** such as a host workstation is provided to load configuration data into the emulation system **5**.

The emulation system **5** is usually constructed from individual configurable logic devices **12**, specifically FPGA chips are common. The configurable logic devices **12** are connected to each other via an interconnect **14**. Memory elements **6** are also optionally provided and are accessible by the configurable logic devices **12** through the interconnect **14**.

The host workstation **2** downloads the configuration data that will dictate the internal configuration of the logic devices **12**. The configuration data is compiled from a digital circuit description that includes the desired manner in which the emulation system **5** is intended to interact with the environment or target system **4**. Typically, the target system **4** is a larger electronic system for which some component such as a microprocessor is being designed. The description applies to this microprocessor and the emulation system **5** loaded with the configuration data confirms compatibility between the microprocessor design and the target system **4**. Alternatively, the target system **4** can be a device for which the logic system satisfies some processing requirements. Further, the emulation system **5** can be used for simulations in a software or FPGA based logic simulation.

FIG. **2** illustrates how the logic design is distributed among the logic devices **12** of the logic system **5**. A netlist **20** describing the logic connectivity of the logic design is separated into logic partition blocks **22**. The complexity of the blocks **22** is manipulated so that each can be realized in a single FPGA chip **12**. The logic signal connections that must bridge the partition blocks **24**, global links, are provided by the interconnect **14**. Obviously, the exemplary netlist **20** has been substantially simplified for the purposes of this illustration.

FIG. **3** illustrates the origins of hold time problems in conventional logic designs. The description is presented in the specific context of a configurable system **100**, such as an emulation system, comprising four configurable logic devices **110**–**116**, such as FPGAs, which are interconnected via a crossbar **120** interconnect. A portion of the internal logic of these devices is shown to illustrate the distribution of a gated clock and the potential problems from the delay of the clock.

The second logic device **112** has been programmed with a partition of the intended logic design that includes an edge-triggered D-type flip-flop **122**. This flip-flop **122** receives a data signal DATA at an input terminal D1 and is clocked by a clock signal CLK, both of which are from the environment in which the system **100** is intended to operate. The output terminal Q1 of the first flip-flop is connected to a second flip-flop **124** in the fourth logic device **116** through the crossbar **120**. This second flip-flop **124** is also clocked by the clock signal, albeit a gated version that reaches the second flip-flop **124** through the crossbar **120**, through combinational logic **126** on a third configurable logic device **114** and through the crossbar **120** a second time before it reaches the clock input of the second flip-flop **124**.

Ideally, the rising edge of the clock signal should arrive at both the first flip-flop **122** and the second flip-flop **124** at precisely the same time. As a result of this operation, the logic value "b" held at the output terminal Q1 of the first flip-flop **122** and appearing at the input terminal D2 of the second flip-flop **124** will be latched to the output terminal Q2 of the second flip-flop **124** as the data input is latched by flip-flop **122**. The output terminals Q1 and Q2 of the flip-flops **122**, **124** will now hold the new output values "a" and "b". This operation represents correct synchronous behavior.

The more realistic scenario, especially when gated clocks are used, is that the clock signal CLK will not reach both of the flip-flops **122** and **124** at the same instant in time. This

realistic assumption is especially valid in the illustrated example in which the clock signal CLK must pass through the combinational logic **126** on the third configurable logic device **114** before it reaches the second flip-flop **124** on the fourth configurable logic device **116**. In this example, assume the clock signal CLK reaches the first flip-flop **122** in the second configurable logic device **112** and clocks the value at that flip-flop's input terminal D**1** to the output Q**1**. At some point, the output Q**1** of the first flip-flop is now holding the new value "a" and this new value begins to propagate toward the input D**2** of the second flip-flop **124**. The rising edge of the clock signal CLK has not propagated to the second flip-flop **124** on the fourth configurable logic device **116**, however. Instead, a race of sorts is established between the rising edge of the clock signal CLK and the new value "a" to the second flip-flop **124**. If the new value "a" reaches the input terminal D**2** of the second flip-flop before the rising edge of the clock signal CLK, the old value "b" will be over-written. This is incorrect behavior since the information contained in "b" is lost. For correct operation of the circuit, it was required that signal "b" at the input terminal D**2** of the second flip-flop **124** be held valid for a brief period of time after the arrival of the clock edge to satisfy a hold time requirement. Unfortunately, unpredictable routing and logic delays postpone the clock edge beyond the validity period for the input signal "b".

In environments where delays can not be predicted precisely, hold time violations are a serious problem that can not be rectified merely by stretching the length of the clock period. Often, there is a need for careful delay tuning in traditional systems, either manually or automatically, in which analog delays are added to signal paths in the logic. The delays usually require further decreases in the operational speed of the target system. This lengthens the periods of the environmental timing signals and gives the emulation system more time to perform the logic calculations. These changes, however, create their own timing problems, and further erode the overall speed, ease-of-use, and predictability of the system.

FIG. **4**A is a schematic diagram showing the internal architecture of the logic system **200** which has been configured according to the principles of the present invention. This logic system **200** comprises a plurality of configurable logic devices **214***a*–**214***d*. This, however, is not a strict necessity for the invention. Instead, the logic system **200** could also be constructed from a single logic device or alternatively from more than the four logic devices actually shown. The logic devices are shown as being connected by a Manhattan style interconnect **418**. Again, the interconnect is non-critical, modified Manhattan-style, crossbars or hierarchial interconnects are other possible and equivalent alternatives.

The internal logic architecture of each configurable logic device **214***a*–**214***d* comprises a finite state machine **428**–**434** and logic **420**–**426**. An internal or virtual clock VClk generates an internal or virtual clock signal that is distributed through the interconnect **418** to each logic device **214***a*–**214***d*, and specifically, the logic **420**–**426** and finite state machines **428**–**434**. Generally, the logic **420**–**426** performs the logic operations and state transitions associated with the logic design that was developed from the digital circuit description. The finite state machines **428**–**434** control the sequential operations of the logic in response to the signal from the virtual clock VClk.

The logic system **200** operates synchronously with the single internal clock signal VClk. Therefore, a first synchronizer SYNC**1** and a second synchronizer SYNC**2** are provided to essentially generate synchronous versions of timing signals from the environment. In the illustrated example, they receive environmental timing signals EClk**1** and EClk**2**, respectively. The synchronizers SYNC**1** and SYNC**2** also receive the internal clock signal VClk. Each of the synchronizers SYNC**1** and SYNC**2** generates a synchronizing control signal V_{G01}, V_{G02} in response to an edge of the respective environmental timing signal EClk**1** and EClk**2**, upon the next transition of the internal clock VClk. Thus, these control signals are synchronous with the internal clock.

FIG. **4**B shows an exemplary timing diagram of the virtual clock signal VClk compared with a first environmental clock signal EClk**1** and a second environmental clock signal EClk**2**. As shown, typically, the virtual clock VClk is substantially faster than any of the environmental clocks, at least four times faster but usually faster by a factor of ten to twenty. As a general rule, the temporal resolution of the virtual clock, i.e., the cycle time or period of the virtual clock, should be smaller than the time difference between any pair of environmental timing signal edges.

In the example, the environmental clocks EClk**1** and EClk**2** are rising edge-active. The signals V_{G01} and V_{G02} from the first synchronizer SYNC**1** and the second synchronizer SYNC**2**, respectively, are versions of the environmental clock which are synchronized to the internal clock VClk in duration. The transitions occur after the rising edges of the environmental clocks EClk**1** and EClk**2**, upon the next or a later rising edge of the internal clock. For example, the second synchronizing signal V_{G02} is active in response to the receipt of the second environmental clock signal EClk**2** upon the next rising edge of the internal clock VClk.

Returning to FIG. **4**A, in typical simulation or emulation configurable systems and the present invention, logic of the configurable devices include a number of interconnected combinational components that perform the boolean functions dictated by the digital circuit design. An example of such components are logic gates. Other logic is configured as sequential components. Sequential components have an output that is a function of the input and state and are clocked by a timing signal. An example of such sequential components would be a flip-flop. In the typical configurable systems, the environmental clock signals are provided to the logic in each configurable logic device to control sequential components in the logic. This architecture is a product of the emulated digital circuit design in which similar components were also clocked by these timing signals. The present invention, however, is configured so that each one of these sequential components in the logic sections **420**–**426** is clocked by the internal or virtual clock signal VClk. This control is schematically shown by the distribution of the internal clock signal VClk to each of the logic sections **420**–**426** of the configurable devices **410**–**416**. As described below, the internal clock is the sole clock applied to the sequential components in the logic sections **420**–**426** and this clock is preferably never gated.

Finite state machines **428**–**434** receive both the internal clock signal VClk and also the synchronizing signals V_{G01} V_{G02} from the synchronizers SYNC**1** and SYNC**2**. The finite state machines **428**–**434** of each of the configurable logic devices **410**–**416** generate control signals to the logic sections **420**–**426**. These signals control the operation of the sequential logic components. Usually, the control signals are received at load enable terminals. As a result, the inherent functionality of the original digital circuit design is maintained. The sequential components of the logic are operated in response to environmental timing signals by virtue of the

fact that loading occurs in response to the synchronized versions of the timing signals, i.e. V_{G01} V_{G02}. Synchronous operation is imposed, however, since the sequential components are actually clocked by the single internal clock signal VClk throughout the logic system **200**. In contrast, the typical simulation or emulation configurable systems would clock the sequential components with the same environmental clock signals as in the original digital circuit description.

It should be noted that separate finite state machines are not required for each configurable logic device. Alternatively, a single finite state machine having the combined functionality of finite state machines **428–434** could be implemented. For example, one configurable device could be entirely dedicated to this combined finite state machine. Generally, however, at least one finite state machine on each device chip is preferred. The high cost of interconnect bandwidth compared to on-chip bandwidth makes it desirable to distribute only the synchronizing signals V_{G01} V_{G02} to each chip, and generate the multiple control signals on-chip to preserve the interconnect for other signal transmission.

FIG. **5**A shows a portion of a logic circuit that has been programmed into the logic system **200** according to the present invention. This logic circuit is a resynthesized version of the logic circuit shown in FIG. **3**. That is, the logic circuit of FIG. **5**A and of FIG. **3** have many of the same characteristics. Both comprise flip-flops **122** and **124**. The flip-flop **122** has an output terminal Q**1** which connects to the input terminal D**2** of flip-flop **124**. Further, the combinational logic **126** is found in both circuits.

The logic circuit of FIG. **5**A differs from FIG. **3** first in that each of the flip-flops **122** and **124** are load-enable type flip-flops and clocked by a single internal clock VClk. Also, the environmental clock signal Clk is not distributed per se to both of the flip-flops **122** and **124** as in the circuit of FIG. **3**. Instead, a synchronized version of the clock signal V_{G0} is distributed to a finite state machine **430** of the second configurable logic device **214***b* and is also distributed to a finite state machine **434** of the fourth configurable logic device **214***d*. The finite state machine **430** then provides a control signal to a load enable terminal LE**1** of flip-flop **122** and finite state machine **434** provides a control signal to the load enable terminal LE**2** of flip-flop **124** through the combinational logic **126**.

FIG. **5**B is a timing diagram showing the timing of the signals in the circuit of FIG. **5**A. That is, at time **510**, new data is provided at the input terminal D**1** of flip-flop **522**. Then, at some later time, **512**, the clock signal Clk is provided to enable the flip-flop **122** to clock in this new data. The second flip-flop **124** is also intended to respond to the environmental clock signal Clk by capturing the previous output of flip-flop **122** before that flip-flop is updated with the new data. Recall that the problem in the logic circuit of FIG. **3** was that the clock signal to the second flip-flop **124** was gated by the combinational logic **126** which delayed that clock signal beyond time at which the output "b" from the output terminal Q**1** of the flip-flop **122** was valid. In the present invention, the environmental clock signal Clk is received at the synchronizer SYNC. This synchronizer also receives the virtual clock signal VClk. The output of the synchronizer V_{G0} is essentially the version of the environmental clock signal that is synchronized to the internal clock signal. Specifically, the new signal V_{G0} has rising and falling edges that correspond to the rising edges of the internal clock signal VClk.

The finite state machines **430** and **434** are individually designed to control the flip-flops in the respective configurable logic **214***b* and **214***d* to function as required for correct synchronous operation. Specifically, finite state machine **434** generates a control signal **215** which propagates through the combinational logic **126** to the load enable terminal LE**2** of the flip-flop **124**. This propagation of control signal **215** from finite state machine, through combinational logic **126**, to LE**2** occurs in a single virtual clock cycle. The generation of control signal **215** precedes the generation of control signal **217** by the finite state machine **430** by a time of two periods (for example) of the internal clock VClk. This two cycle difference, **514**, assumes that flip-flop **124** is enabled before flip-flop **122** is enabled, thereby latching "b", and thus providing correct operation. As a result, both flip-flop **122** and flip-flop **124** are load enabled in a sequence that guarantees that a new value in flip-flop **122** does not reach flip-flop **124** before flip-flop **124** is enabled. In fact, if the compiler has scheduled "b" to arrive at D**2** on some cycle, x, later than **217**, then the compiler can cause control signal **215** to be available on that cycle x, or later. In the above instance, the correct circuit semantics is preserved even though control signal **215** arrives after control signal **217**. The key is that **215** must enable flip-flop **124** in a virtual cycle in which "b" is at D**2**.

Further, the precise control of storage elements afforded by the present invention allows set up and hold times into the target system to be dictated. In FIG. **5**A, output Q**2** of flip-flop **124** is linked to a target system via a third flip-flop **140**. The flip-flop **140** is load enabled under the control of finite state machine **434** and clocked by the virtual clock. Thus, by properly constructing this finite state machine **434**, the time for which flip-flop **140** holds a value at terminal Q**3** is controllable to the temporal resolution of a cycle or period of the virtual clock signal.

This aspect of the invention enables the user to test best case and worst case situations for signal transmission to the target system and thereby ensure that the target system properly captures these signals. In a similar vein, this control also allows the user to control the precise time of sampling signals from the target system by properly connected storage devices.

FIG. **6** illustrates a method by which a digital circuit design with an arbitrary clocking methodology and state elements is transformed into a new circuit that is synchronous with the internal clock signal but is a functional equivalent of the original digital circuit. The state elements of the new circuit are exclusively edge triggered flip-flops.

The first step is specification **610**. This is a process by which the digital circuit design along with all of the inherent timing methodology information required to precisely define the circuit functionality is identified. This information is expressed in four pieces, a first piece of which is the gate-level circuit netlist **610***a*. This specifies the components from which the digital circuit is constructed and the precise interconnectivity of the components.

The second part **610***b* of the specification step **610** is the generation of a functional description of each component in the digital circuit at the logic level. For combinatorial components, this is a specification of each output as a boolean function of one or more inputs. For example, the specification of a three input OR gate—inputs A, B, and C and an output O—is O=A+B+C. For sequential components, this entails the specification of outputs as a boolean function of the inputs and state. The specification of the new state as a boolean function of the inputs and state is also required for the sequential components along with the specification of when state transitions occur as a function of either boolean

inputs or directed input transitions. A directed input transition is a rising or falling edge of an input signal, usually a timing signal from the environment in which the logic system **200** is intended to ultimately function. For example, the specification of a rising edge-triggered flip-flop—inputs D, CLK, of output Q, and state S—is Q=S, S=D, and state transition when CLK rises.

Another part of the specification step is the description of the timing relationships of the inputs to the logic system step **610***c*. This includes environment timing signals and environmental input signals and the relationship to the output signals generated by the logic system **200** to the environment. Input signals to the logic system **200** can be divided into two classes: timing signals and environmental data signals. The timing signals are generally environmental clock signals, but can also be asynchronous resets and any other form of asynchronous signal that combinatorially reach inputs of state elements involved in the functions triggering state transitions. In contrast, environmental data signals include environmental output signals and output signals to the environment that do not combinatorially reach transition controlling inputs of state elements. The timing relationship also specify the timing of environmental data signals relative to a timing signal.

The specification step must also include the specification of the relative timing relationships for all timing signals step **610***d*. These relationships can be one of three types:

A basket of timing signals can be phase-locked. Two signals of equal frequency are phase-locked if there is a known phase relationship between each edge of one signal and each edge of the other signal. For example, the first environmental clock signal and the second environmental clock signal illustrated in FIG. **4** would be phase-locked signals. Additionally, two signals of integrally related frequency are phase-locked if there is a known phase relationship, relative to the slower signal, between any edge of the slower signal and each edge of the faster signal. Two signals of rationally related frequency are phase-locked if they each are phase-locked to the same slower signal.

Another type of timing relationship is non-simultaneous. Two signals are non-simultaneous if a directed transition in one signal guarantees that no directed transition will occur in the other within a window around the transition of some specified finite duration. If two signals are non-simultaneous and also not phase-locked, this implies that one signal is turned off while the other is on and vice versa. For example, two non-simultaneous signals might be two signals that indicate the mutually exclusive state of some component in the environment. The first signal would indicate if the component was in a first condition and the second timing signal would indicate if the component were in a second condition and the first and second condition could never happen at the same time.

Finally, the last type of relationship is asynchronous. Two signals are asynchronous if the knowledge about a directed transition of one of the signals imparts no information as to occurrence of a transition in the other signal.

It should be recognized that phase-locked is a transitive relationship so that there will be collections of one or more clocks that are mutually phase-locked with respect to each other. Such collection of phase-locked clocks is referred to as a domain. Relationship between domains are either non-simultaneous or asynchronous. The timing signals must be decomposed into a collection of phase-locked domains, and the relationship between pairs of the resulting domains, either synchronous or non-simultaneous, must be specified.

The ordering of the edges of timing signals within each domain are also specified. For example, first CLK**1** rises, then CLK**2** rises, then CLK**2** falls and then CLK**1** falls.

A transition analysis step **612**, value analysis step **614**, and sampling analysis step **616** are used to determine when, relative to the times at which transitions occur on timing signals, signals within a digital circuit change value, and where possible, what these values are. Also determined is when the values of particular signals are sampled by state elements within the circuit as a separate analysis.

In the transition analysis step **612**, a discrete time range is established for each clock domain including one time point for each edge of a clock within the domain. All edges within the domain are ordered and the ordering of time points corresponds to this ordering of edges.

In the value analysis step **614**, the steady state characteristics of every wire in the digital circuit is determined for each discrete time range. Within a discrete time range, any wire within the digital circuit can either be known to be 0, known to be 1, known not to rise, known not to fall or known not to change, or a combination of not falling and not rising. A conservative estimate of the behavior of an output of a logic component can be deduced from the behavior of its inputs. Information about environmental timing signals and environmental data signals can be used to define their behavior. Based on the transition and value information of the inputs to the logic system corresponding information can be deduced for the outputs of each component. A relaxation algorithm is used, in which output values of a given component are recomputed any time an input changes. If the outputs in turn change, this information is propagated to all the places the output connects, since these represent more inputs which have changed. The process continues until no further changes occur.

A second relaxation process, similar to that for transition and value analysis, is used in the sampling step **616**. Sampling information reflects the fact that at some point in time, the value carried on a wire may be sampled by a state element, either within the logic system **200** or by the environment. Timing information for output data signals to the environment provides an external boundary condition for this relaxation process. Additionally, once transition analysis has occurred, it is possible to characterize when all internal state elements potentially make transitions and thus when they may sample internal wires. Just as with transition and value propagation, the result is a relationship between inputs and outputs of a component. For sampling analysis, it is possible to deduce the sampling behavior of inputs of a component from the sampling information for its outputs. The relaxation process for computing sampling information thus propagates in the opposite direction from that of transition information, but otherwise similarly starts with boundary information and propagates changes until no further changes occur.

At the termination of transition **612** and sampling **616** steps it is possible to characterize precisely which timing edges can result in transitions and/or sampling for each wire within the digital circuit. Signals which are combinationally derived from timing signals with known values often also carry knowledge about their precise values during some or all of the discrete time range. They similarly often are known to only be able to make one form of directed transition, either rising or falling, at some particular discrete time point. This information is relevant to understanding the behavior of edge-triggered state elements.

The final resynthesis step **618** involves the application of a number of circuit transformations to the original digital

circuit design which have a number of effects. First, the internal clock VClk is introduced into the logic design **200** of the digital circuit. The internal clock signal is the main clock of the logic system **200**. Further, in effect, all of the original environmental timing signals of the digital circuit are converted into data signals in the logic system **200**. Finally, all of the state elements in the digital circuit are converted to use the internal clock signal as their clock, leaving the internal clock as the only clock signal of the transformed system. The state elements of the original digital circuit design are converted preferably into edge-triggered flip-flops and finite state machines, which generate control signals to the load enable terminals of the flip-flops. The information developed in the transition analysis step **612**, value analysis step **614**, and sampling analysis step **616** is used to define the operation of the finite state machines as it relates to the control of the flip-flops in response to the internal clock signal and the environmental timing signals. The finite state machines send load enable signals to the flip-flops when it is known that data inputs are correct based upon a routing and scheduling algorithm described in the U.S. patent application Ser. No. 08/344,723 filed Nov. 23, 1994 and entitled "Pipe-Lined Static Router and Scheduler for Configurable Logic System Performing Simultaneous Communications and Computations", incorporated herein by this reference. The scheduling algorithm essentially produces a load enable signal on a virtual clock cycle that is given by the maximum of the sum of data, value available time, and routing delays for each signal that can affect data input.

Single Flip-Flop Timing Resynthesis

FIG. **7**A shows a simple edge-triggered flip-flop **710** which was a state element in the original digital circuit. Specifically, the edge-triggered flip-flop **710** receives some input signal at its input terminal D and some timing signal, such as an environmental clock signal ECLK at its clock input terminal. In response to a rising edge received into this clock terminal, the value held at the input terminal D is placed at the output terminal Q.

The timing resynthesis step converts this simple edge-triggered flip-flop **710** to the circuit shown in FIG. **7**B. The new flip-flop is a load-enabled flip-flop and is clocked by the internal clock signal VClk. The enable signal of the converted flip-flop is generated by a finite state machine FSM. Specifically, the finite state machine monitors a synchronized version of the clock signal V_{G0} and asserts the enable signal to the enable input terminal E of the converted flip-flop **720** for exactly one cycle of the internal clock VClk in response to synchronizing signal V_{G0} transitions from 0 to 1. The finite state machine is programmed so that the enable signal is asserted on an internal clock signal cycle when the input IN is valid accounting for delays in the circuit that arise out of a need to route the signal IN on several VClk cycles from the place it is generated to its destination at the input of flip-flop **720**. In a virtual wire systems signals are routed among multiple FPGAs on specific internal clock VClk cycles. The synchronizing signal V_{G0} is generated by a synchronizer SYNC in response to receiving the environmental timing signal EClk on the next or a following transition of the internal clock signal VClk. As a result, the circuit is functionally equivalent to the original circuit shown in FIG. **7**A since the generation of the enable signal occurs in response to the environmental clock signal EClk each time a transition occurs. The circuit, however, is synchronous with the internal clock VClk.

In a digital circuit comprising combinational logic and a collection of flip-flops, all of which trigger off the same edge of a single clock, the basic timing resynthesis transformation, shown in FIG. **7**B and described above, can be extended. All flip-flops are converted to load-enabled flip-flops and have their clock inputs connected to the internal clock VClk. The load enable terminal E of each flip-flop is connected to enable signals generated by a shared finite state machine in an identical manner as illustrated above. The FSM can be distinct for each FPGA. The enables for each flip-flop will be produced to account for routing delays associated with each signal input to the flip-flops.

Timing Resynthesis for Domains for Multiple Clocks

FIG. **8**A shows a circuit comprising three flip-flops **810–814** that are clocked by two environmental clock signals EClk**0** and EClk**1**. For the purposes of this description, both environmental clock signals EClk**0** and EClk**1** are assumed to be phase-locked with respect to each other.

The transformed circuit is shown in FIG. **8**B. It should be noted that the basic methodology of the transform is the same as described in relation to FIGS. **7**A and **7**B. The finite state machine FSM and the clock sampling circuitry SYNC**1** and SYNC**2** have been extended. As before, each flip-flop of the transformed circuit has been replaced with a load-enabled positive-edge triggered flip-flop **820–824** in the transformed circuit. The first environmental clock signal EClk**0** and the second environmental clock signal EClk**1** are synchronized to the internal clock by the first synchronizer SYNC**0** and the second synchronizer SYNC**1**. The synchronizing signals V_{G00} and V_{G01} are generated by the synchronizers SYNC**0** and SYNC**1** to the finite state machine FSM. The finite state machine FSM watches for the synchronizing signals V_{G00} and V_{G01} and then produces a distinct load enable pulse C**0**-Rise, C**0**-Fall, C**1**-Rise for each timing edge on which the clocks EClk**0** and EClk**1** of the flip-flops **820–824** operate. The ordering of these load enable pulses is prespecified within a domain where there is a unique ordering of the edges of all phase-locked clocks. This unique ordering of clocks is specified by the user of the system. As with the single clock case shown in FIG. **7**B, each of the enable pulses C**0**-Rise, C**0**-Fall, and C**1**-Rise is asserted for exactly one period of the internal clock VClk upon detection of the corresponding clock edge in FIG. **8**B.

Multiple Clock Domains Resynthesis

FIG. **9**A shows a collection of flip-flops **910–912** from the digital circuit having multiple clock domains. That is, the first clock signal CLK**0** and the second clock signal CLK**1** do not have a phase-locked relationship to each other, rather the clocks are asynchronous with respect to each other.

FIG. **9**B shows the transformed circuit. A different finite state machine FSM**0** and FSM**1** is assigned to each domain. specifically, a first finite state machine FSM**0** is synchronized to the first environmental clock EClk**0** to generate the load enable signal to the load enable terminal E**0** of the first flip-flop **920**. The second finite state machine FSM**1** generates a load enable signal to E**1** of the second flip-flop **922** in response to the second environmental clock signal EClk**1**. It should be noted, however, that although FSM**0** and FSM**1** operate independently of each other, each of whose sequences are initiated by separate signals V_{g00} and V_{g01}, and that although the first flip-flop **920** and the second flip-flop **922** work independently of each other, i.e., load enabled by different clock signals EClk**0** and EClk**1**, the resulting system is a single-clock synchronous system with the internal clock VClk.

The relationship between the behavior of the first finite state machine FSM**0** and the second finite state machine FSM**1** of the two clock signal domains is related to the relationship between the domains themselves. When the two domains are asynchronous, the first finite state machine FSM**0** and the second finite state machine FSM**1** may operate simultaneously or non-simultaneously. When the two domains are non-overlapping, the first finite state machine FSM**0** and the second finite state machine FSM**1** never operate simultaneously since the edges within the domains are separated in time.

The simultaneity of operation of finite state machines that are asynchronous with respect to each other leaves two circuits which can not readily be transformed by timing resynthesis. A state element which can undergo transitions as a result of an edge produced from a combination of signals in asynchronously related domains can not be resynthesized. Such condition can rise if two asynchronous clocks are gated together and fed into the clock input of a flip-flop or if a state element with multiple clocks and/or asynchronous presets or clears is used as transition triggering inputs from distinct asynchronously related clock domains. Due to the non-simultaneous events and non-overlapping domains, the situations above are not problematic in the non-overlapping situation.

Gated Clock Transformations

Clock gating in the digital circuit provides additional control over the behavior of state elements by using combinational logic to compute the input to clock terminals. The timing resynthesis process transforms gated clock structures into functionally equivalent circuitry which has no clock gating. Generally, gated clock structures can be divided into two classes: simple gated clocks and complex gated clocks. The basis for this distinction lies in the behavior of the gated clocks as deduced from timing analysis. Previously, the terms timing signal and data signal were defined in the context of inputs and outputs to the digital circuit. A gated clock is a combinational function of both timing signals and data signals. The gated clock transition then controls the input of a state element. Data signals can either be external input data signals from the environment or internally generated data signals.

A simple gated clock has two properties:

1) at any discrete time it is possible for a simple gated clock to make a transition in at most one direction, stated differently, there is no discrete time at which the simple gated clock may sometime rise and sometime fall; and

2) only timing signals change at those discrete times at which state elements can change state.

A complex gated clock violates one of these two properties.

Simple Gated Clock Transformation

FIG. **10**A shows a circuit that exhibits a simple gated clock behavior. FIG. **10**B is a timing diagram showing transitions in the data signal and the gated clock signal as a function of the environmental clock signal EClk. Specifically, upon the falling edge of the environmental clock EClk, the gating flip-flop **1010** latches the control signal CTL received at its input D**1** at its output terminal Q**1**. This is the data signal. The AND gate **1012** receives both the data signal and the environmental clock EClk. As a result, only when the environmental clock EClk goes high, does the gated-clock signal go high on the assumption that the data signal is also a logic high. Upon the rising edge of the gated clock, the second flip-flop **1014** places the input signal IN received at its D**2** terminal to its output terminal Q**2**.

FIG. **10**C shows the transformed circuit. Here, a finite state machine FSM receives a signal V_{G0} from the synchronizer SYNC upon receipt of the environmental clock EClk. The finite state machine FSM produces two output signals: C**0**-Fall which is active upon the falling edge of the environmental clock signal, and C**0**-Rise which is active upon the rising of the environmental clock signal EClk.

The transformed circuit functions as follows. On the first period of the internal clock VClk after the falling edge of the environmental clock signal EClk, the first flip-flop **1016** places the value of the control signal received at its input terminal D**1** to its output terminal Q**1** upon the clocking of the internal clock signal VClk. This output of the first flip-flop **1016** appearing at terminal Q**1** corresponds to the data signal in the original circuit. This data signal is then combined in an AND gate **1020** with the signal C**0**-Rise from the finite state machine FSM that is indicative of the rising edge of the environmental clock signal EClk. The output of the AND gate goes to the load enable terminal E**2** of a second flip-flop **1018** which receives signal IN at its input terminal D**2**. Again, upon the receipt of this load enable and upon the next cycle of the internal clock VClk, the second flip-flop moves the value at its input terminal D**2** to its output terminal Q**2**.

Complex Gated Clock Transformations

In the case of complex gated clock behavior, the factoring technique used for simple gated clock transformations is inadequate. Because data and clocks change simultaneously and/or the direction of a transition is not guaranteed, both the value of a gated clock prior to the transition time and the value of the gated clock after the transition time are needed. Using these two values, it can be determined whether a signal transition that should trigger a state change has occurred. One way to produce the post-transition value of data signals is to replicate the logic computing the signal and also replicate any flip-flops containing values from which the signal is computed and which may change state as a result of the transition. These replica flip-flops can be enabled with an early version of the control signal, thus causing them to take on a new state prior to the main transition. By this mechanism, pre- and post-transition values for signals needed for gated clocks can be produced.

An alternative way to get the two required values for the gated clock signal is to add a flip-flop to record the pre-transition state of the gated-clock and delay in time the update of the state element dependent on the gated clock. These two techniques have different overhead costs and the latter is only applicable if the output of the state element receiving the gated clock is not sampled at the time of the transition. The former always works but the latter generally has lower overhead when applicable.

FIG. **11**A shows two cascaded edge-triggered flip-flops **1100** and **1112**. This configuration is generally known as a frequency divider. The environmental clock signal EClk is received at the clock terminal of the first flip-flop **1110**; and at the output Q**2** of the second flip-flop **1112**, a new clock signal is generated that has one-fourth the frequency of EClk. The divider of FIG. **11**A operates as follows: In an initial state in which the output terminal Q**1** of the first flip-flop **1110** is a 0 and the input terminal D**1** of the flip-flop **1110** is a 1, receipt of the rising edge of the environmental timing signal EClk changes Q**1** to a 1 and D**1** converts to a

0. The conversion of Q1 from 0 to 1 functions as a gated clock to the clock input terminal of the second flip-flop **1112**. The second flip-flop **1112** functions similarly, but since it is only clocked when Q1 of the first flip-flop **1110** changes from 0 to 1, but not 1 to 0, it changes with one-fourth the frequency of EClk.

FIG. **11**B shows the transformed circuit of FIG. **11**A. Here, a replica flip-flop **1120** has been added that essentially mimics the operation of the first flip-flop **1122**. The replica flip-flop **1120**, however, receives a pre-Clk-Rise control signal from the finite state machine FSM. More specifically, the finite state machine FSM responds to the synchronizing signal V_{G0} and the internal clock VClk and produces a pre-CLK-rise signal that is active just prior to the CLK-Rise signal, CLK-Rise being active in response to the rising edge of the environmental timing signal EClk. Assume the output terminal Q1 of the first flip-flop **1122** is initially at a 0 and the input terminal D1 of first flip-flop **1122** is a 1, the replica flip-flop **1120** is initially at a 0. Upon receipt of the pre-CLK-rise signal at the replica flip-flop load enable terminal ER, the output terminal QR of the replica flip-flop **1120** makes a transition from a 0 to a 1. Since Q1 is low and QR is high, an AND gate **1124** functioning as an edge detector generates a high signal. When the CLK-rise control signal from the finite state machine FSM is active in response to receipt of the rising edge of the environmental clock signal EClk, the output terminal Q1 of the first flip-flop **1122** is converted from a 0 to a 1. The enable terminal E2 of flip-flop **1126** also is high, causing the flip-flop to change state. On the next falling transition of Q1, the AND gate **1124** will produce 0 and flip-flop **1126** will not change state. Since the replica flip-flop **1120** provides a zero to the rising edge detector whenever the zero is present at the input terminal of the first flip-flop, the rising edge detector is enabled only every other transition of Q1.

Latch Resynthesis

Generally, latches are distinguished from flip-flops in that flip-flops are edge-triggered. That is, in response to receiving either a rising or falling edge of a clock signal, the flip-flop changes state. In contradistinction, a latch has two states. In an open state, the input signal received at a D terminal is simply transferred to an output terminal Q. In short, in an open condition, the output follows the input like a simple wire. When the latch is closed, the state of the output terminal Q is maintained or held independent of the input value at terminal D. A semantic characterization of such a latch is as follows. For an input D, output Q, a gate G, and a state S, Q=S. S=D if G=1. The latch is open when G=1 and closed when G=0.

Beginning with the simplest case, if the output of a latch is never sampled when the latch is closed, G=0, the latch is really just a wire. Latches with this characteristic may be used to provide extra hold time for a signal. For this sample latch, this would be true, if the set of discrete times at which the output of the latch is sampled, is equal to or a proper subset of the set of discrete times at which the gate signal G is known to have a value of 1. In this situation, the latch can be removed and replaced with a wire connecting the input and output signals.

In contrast, if the output of the latch is never sampled when the latch is open, the latch is equivalent to a flip-flop. The only value produced by the latch which is ever sampled is a value of the input D on the gate signal edge when the latch transitions from open to closed. This condition is true if the set of discrete times at which the output of the latch is sampled, is equal to, or a proper subset of the discrete times at which the gate signal G is known to have a value of 0. In this situation, the latch can be removed and replaced with an edge-triggered flip-flop.

As shown in FIG. **12**, latches that are open when their gate signal G is high **1210** are converted to negative-edge triggered flip-flops **1212**. Latches that are open when their gate signal G is low **1214** are converted to positive edge triggered flip-flops **1216**.

Once the transition from the latch to the edge triggered flip-flop has been made, these new edge-triggered flip-flops are then further resynthesized by the timing resynthesis techniques described in connection with FIGS. **6–11**. Therefore, after this further processing, both positive and negative edge-triggered flip-flops will be flip-flops clocked by the internal clock VClk. The resynthesized flip-flops will have an enable signal that is generated by a finite state machine in response to the particular environmental clock signal that gated the original latch element.

Referring to FIG. **13**, in the condition in which the output of a given latch **1310** is sampled both when the latch might be open and might be closed, that latch can be converted to a flip-flop **1312**, plus a multiplexor **1314** as shown in FIG. **13**. There, when the gate signal G is low, the multiplexor **1314** selects the input signal to the input terminal D of the flip-flop **1312**. On the rising edge of the gating signal, however, the input to the D terminal is latched at the output terminal Q. Also, at this point, the gating signal selects the second input to the multiplexor **1214**. As with the case in FIG. **12**, the result of the transformation in FIG. **13** is subjected to further resynthesis.

The transform of FIG. **13** may exhibit timing problems if the multiplexor is implemented in a technology that exhibits hazards, or output glitches. Output glitches can and could result in set up and hold time problems of the sampling state element. This transformation can therefore only be used when the output is never sampled at discrete times at which the clock may exhibit an edge. If the output is sampled both when the latch might be opened and closed and some sampling occurs on the edge of the gate signal, a final transformation is employed. A new clock signal is created which is phase-locked to the original clock signal and precedes it.

As shown in FIGS. **14**A and **14**B, the latch **1410** of FIG. **14**A is replaced by a flip-flop which receives the phase-advanced clock indicated by the negative delay **1412** as shown in FIG. **14**B. The state transition of the new flip-flop **1414** precedes a state transition of any circuits sampling the original output Q of the original latch **1410**. If the latch is also sampled when it is open by signals occurring prior to the sampling edge, one of the prior techniques can be employed, either latch to wire or latch to flip-flop and multiplexor transforms of FIG. **13**.

As shown in FIG. **14**B, the negative delay **1412** represents a time-advanced copy of the clock CLK which is used to clock the flip-flop **1414**. While negative-delays are unphysical, this structure can be processed by the timing resynthesis process with a distinct control signal generated by a finite state machine.

FIG. **15** shows a finite state machine FSM generating a pre-CLK-rise control signal one or more cycles of the internal clock VClk prior to the generation of the control signal, CLK-Rise. The control signal CLK-rise is generated in response to the rising edge of the environmental timing signal EClk. As a result, the input signal appearing at the D terminal of the flip-flop **1510** is transferred to the output

terminal prior to the rising edge of the environmental clock signal EClk as signaled by the Clk-rise control signal. Subsequent elements can be then load enabled from the CLK-Rise signal generated by the finite state machine FSM. Here again, if the latch of the original digital circuit is sampled both when the latch is opened and closed, a multiplexor can be placed at the output Q of the flip-flop **1510**.

Combinational Loop Transformations

Combinational loops with an even number of logic inversions around the loop are an implicit state element. An example is shown in FIG. **16**A, this implicit state can be transformed into an explicit state element which is clocked by the virtual clock VClk by simply choosing a wire **1601** in the loop and inserting a flip-flop **1602** which is clocked by the virtual clock VClk as shown in FIG. **16**B.

The addition of the flip-flop **2602** changes the timing characteristics of the loop. Additional virtual clock cycles are required for the values in the loop to settle into their final states.

Assume in FIG. **16**B that all input values to the loop are ready by some virtual cycle V. In the absence of the flip-flop **1602**, all outputs will become correct and stable after some delay period. With the flip-flop **1602**, it is necessary to wait until the loop stabilizes and then wait for an additional virtual clock period during which the flip-flop value may change and subsequently change the loop outputs. Thus the outputs of the loop cannot be sampled until virtual cycle V+1.

If combinational cycles are nested, each can be broken by the insertion of a flip-flop as above. Nested loops may require up to 2^N clock cycles to settle, where N is the depth of the loop nesting and thus the number of flip-flops needed to break all loops.

RS Latch Transformations

RS latches **1710** are asynchronous state elements built from cross-coupled NOR or NAND gates **1712**, as illustrated in FIG. **17**A.

RS latches **1710** can be transformed based on the transformation for combinational cycles illustrated in FIGS. **16**A and **16**B. An alternative approach illustrated in FIG. **17**B eliminates the combinational cycles associated with RS latches while also avoiding the extended settling time associated with the general combinational cycle transformation of FIG. **16**B.

The circuit is FIG. **17**B forces the outputs Q and $\overline{Q}$ of the RS latch **1710** combinatorially to their values for all input patterns except the one in which the latch maintains its state. For this pattern, the added flip-flop **1714** produces appropriate values on the outputs. Logic **1716** is provided to set the flip-flop **1714** into an appropriate state, based on the values of the inputs whenever an input pattern dictates a state change. When the latch **1710** is maintaining its state, the outputs will be stable so no propagation is required. Thus the outputs of the transformation are available with only a combinatorial delay.

A symmetrical transformation can be applied to latches produced from cross-coupled NOR gates.

Asynchronous Presets and Clears

Asynchronous presets and clears of state elements shown in FIG. **18**A can be transformed in one of two ways. Each transformation relies on the fact that preset and clear signals R are always synchronized to the virtual clock, either because they are internally generated by circuitry which is transformed to be synchronous to the virtual clock or because they are external asynchronous signals which are explicitly synchronized using synchronizer circuitry.

The first transformation, shown in FIG. **18**B, makes use of an asynchronous preset or clear on flip-flop **1808** in the FPGA, if such exists. The enable signal E which enables the resynthesized state element to undergo state changes is used to suppress/defer transitions on the preset or clear input R to eliminate race conditions arising from simultaneously clocking and clearing or presenting a state element.

The second transformation shown in FIG. **18**C converts an asynchronous preset or clear R_V which has already been synchronized to the clock into a synchronous preset or clear. The enable signal E to the resynthesized state element must be modified to be enabled at any time at which a preset or clear transition might occur by gate **1810**.

Returning to FIG. **6**, the above described transformations of the timing resynthesis step **618** in combination of with the specification step **610**, transition analysis **612**, value analysis **614** and sampling analysis **616** enable conversion of a digital circuit description having some arbitrary clocking methodology to a single clock synchronous circuit. The result is a circuit which the state elements are edge-triggered flip-flops. To generate the logic system **200** having the internal architecture shown in FIG. **4**, this resynthesized circuit must now be compiled for and loaded into the configurable logic devices **410–416** by the host workstation **222**.

FIG. **19** shows the complete compilation process performed by the host workstation **222** to translate the digital circuit description into the configuration data received by the configurable devices **214**. More specifically, the input to a compiler running on the host workstation **222** is the digital circuit description in step **1610**. This description is used to generate the resynthesized circuit as described above. The result is a logic netlist of the resynthesized circuit **1611**. This includes the new circuit elements and the new VClk.

In step **1612**, functional simulations of the transformed circuit can be performed. This step ensures that the resynthesized circuit netlist is the functional equivalent of the original digital circuit. It should be noted that the transformed circuit is also more amenable to computer-based simulations. All relevant timing information specifying the behavior of the timing signals including the timing relationship to each other is built into the resynthesized circuit yet the resynthesized circuit is synchronous with a single clock. Therefore, the resynthesized circuit could alternatively be used as the circuit specification for a computer simulation rather than the hardware based simulation on the configurable logic system. The resynthesized circuit is then partitioned **1613** into the logic partition blocks that can fit into the individual FPGAs of the array, see FIG. **2**.

In the preferred embodiment of the present invention, techniques described in U.S. patent application Ser. No. 08/042,151, filed on Apr. 2, 1993, entitled Virtual Wires for Reconfigurable Logic System, which is incorporated herein by this reference, are implemented to better utilize pin resources by multiplexing global link transmission on the pins of the FPGAs across the interconnect. Additionally, as described in incorporated U.S. patent application Ser. No. 08/344,723, filed on Nov. 23, 1994, entitled Pipe-Lined Static Router and Scheduler for Configurable Logic System Performing Simultaneous Communication and Computation, signal routing is scheduled so that logic computation and global link transmission through the interconnect happen simultaneously.

Specifically, because a combinatorial signal may pass through several FPGA partitions as global links during an emulated clock cycle, all signals will not be ready to schedule at the same time. This is best solved by performing a dependency analysis, step **1614** on global links that leave a logic partition block. To determine dependencies, the partition circuit is analyzed by backtracing from partition outputs, either output global links or output signals to the target system, to determine on which partition inputs, either input links or input signals from the target system, the outputs depend. In backtracing, it is assumed that all outputs depend on all inputs for gate library parts, and no outputs depend on any inputs for latch or register library parts. If there are no combinatorial loops that cross partition boundaries, this analysis produces a directed acyclic graph, used by a global router. If there are combinatorial loops, then the loops can be hardwired or implemented in a single FPGA. Loops can also be broken by inserting a flip-flop into the loop and allowing enough virtual cycles for signal values to settle to a stable state in the flip-flop.

Individual FPGA partitions must be placed into specific FPGAs (step **1616**). An ideal placement minimizes system communication, requiring fewer virtual wire cycles to transfer information. A preferred embodiment first makes a random placement followed by cost-reduction swaps and then optimizes with simulated annealing. During global routing (step **1618**), each global link is scheduled to be transferred across the interconnect during a particular period of the pipe-line clock. This step is discussed more completely in the incorporated U.S. patent application Ser. No. 08/344,723, Pipe-Lined Static Router and Scheduler for Configurable Logic System Performing Simultaneous Communication and Computation.

Once global routing is completed, appropriately-sized multiplexors or shift loops, pipeline registers, and associated logic such as the finite state machines that control both the design circuit elements and the multiplexors and pipeline registers are added to each partition to complete the internal configuration of each FPGA chip **22** (steps **1620**). See specifically, incorporated U.S. patent application Ser. No. 08/042,151, Virtual Wires for Reconfigurable Logic System. At this point, there is one netlist for each configurable logic device **214** or FPGA chip. These FPGA netlists are then processed in the vender-specific FPGA place-and-route software (step **1622**) to produce configuration bit streams (step **1624**). Technically, there is no additional hardware support for the multiplexing logic which time-multiplex the global links through the interconnect: the array of configurable logic is itself configured to provide the support. The necessary "hardware" is compiled directly into the configuration of the FPGA chip **214**. Some hardware support in the form of special logic for synchronizers to synchronize the external clocks to the internal VClk is recommended.

While this invention has been particularly shown and describe with references to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the spirit and scope of the invention as defined by the appended claims. For example, it is not a strict necessity that the internal clock signal VClk be distributed directly to the sequential logic elements. Preferably it reaches each element at substantially the same time. In some larger networks, therefore, some delay may be preferable to delay tune the circuit for propagation delays.

We claim:

1. A method of configuring a configurable logic system to operate in an environment, the logic system generating output signals to the environment in response to at least one environmental timing signal and environmental data signals provided from the environment, the method comprising:

configuring the logic system to perform logic operations for generating the output signals in response to the environmental data signals and an internal clock signal; and

configuring the logic system to comprise a finite state machine for generating control signals to control the logic operations in response to the environmental timing signal and the internal clock signal.

2. A method of configuring as described in claim **1**, further comprising configuring the logic system to comprise a synchronizer for sampling the environmental timing signal in response to the internal clock signal.

3. A method of configuring as described in claim **1**, wherein the logic system comprises at least one field programmable gate array.

4. A method of configuring as described in claim **1**, further comprising configuring the finite stare machine to dictate set-up and hold times of signals to the environment.

5. A method of configuring as described in claim **1**, further comprising configuring the finite state machine to dictate sampling times of the environmental data signals.

6. A method of configuring as described in claim **1**, further comprising configuring the logic system to have combinational logic and sequential logic to perform the logic operations.

7. A method of configuring as described in claim **6**, further comprising configuring the finite state machine to generate control signals to the sequential logic in response to the environmental timing signal and the internal clock signal.

8. A method of configuring as described in claim **7**, further comprising configuring the sequential logic to comprise flip-flops receiving the internal clock signal at a clock input and the control signals at a latch enable input.

9. A method of configuring as described in claim **1**, wherein the logic system comprises a plurality of configurable logic devices electrically connected via an interconnect for transmitting signals between the chips.

10. A method of configuring as described in claim **9**, wherein the interconnect comprises cross bar chips.

11. A method as configuring as described in claim **9**, wherein the interconnect utilizes a direct-connect topology.

12. A method of configuring as described in claim **11**, wherein the interconnect includes buses.

13. A logic system for generating output signals to an environment in response to at least one environmental timing signal and environmental data signals provided from the environment, the logic system comprising:

an internal clock for generating an internal clock signal for the logic system;

at least one configurable logic device including:

logic which generates the output signals in response to the environmental data signals and the internal clock signal; and

a finite state machine which coordinates operation of the logic in response to the internal clock signal and the environmental timing signal.

14. A logic system as described in claim **13**, wherein the at least one configurable logic device comprises at least one field programmable gate array.

15. A logic system as described in claim **13**, further comprising an interconnect for transmitting signals between plural configurable logic devices.

16. A logic system as described in claim **13**, further comprising a synchronizer for sampling the environmental timing signal in response to the internal clock signal.

17. A logic system as described in claim **16**, wherein the synchronizer is constructed from non-programmable logic.

18. A logic system as described in claim **13**, wherein the logic comprises combinational logic and sequential logic.

19. A logic system as described in claim **18**, wherein the sequential logic comprises flip-flops receiving the internal clock signal at a clock input and the control signals at a latch enable input.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 6,009,531
DATED : December 28, 1999
INVENTOR(S) : Charles W. Selvidge and Matthew L. Dahl

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

In column 2, line 34, replace "he" with --the--.
In column 2, line 35, replace "he" with --the--.
In column 22, line 24, replace "stare" with --state--.

Signed and Sealed this

Twenty-fifth Day of July, 2000

Attest:

Attesting Officer

Q. TODD DICKINSON

Director of Patents and Trademarks

U.S. Patent 6,947,882

A552 - 572

US006947882B1

(12) **United States Patent**
Reblewski et al.

(10) **Patent No.: US 6,947,882 B1**
(45) **Date of Patent: Sep. 20, 2005**

(54) **REGIONALLY TIME MULTIPLEXED EMULATION SYSTEM**

(75) Inventors: **Frederic Reblewski**, Paris (FR); **Olivier Lepaps**, Paris (FR); **Jean Barbier**, Chatillon (FR)

(73) Assignee: **Mentor Graphics Corporation**, Wilsonville, OR (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/404,920**

(22) Filed: **Sep. 24, 1999**

(51) **Int. Cl.**[7] **G06F 9/455**
(52) **U.S. Cl.** **703/23**; 703/27; 703/24
(58) **Field of Search** 703/28, 23, 24, 703/27; 326/93, 96, 38, 39, 41; 712/15, 37

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,642,487 A		2/1987	Carter	
4,758,985 A		7/1988	Carter	
5,036,473 A		7/1991	Butts et al.	
5,363,319 A	*	11/1994	Okuda	703/15
5,574,388 A		11/1996	Barbier et al.	
5,596,742 A		1/1997	Agarwal et al.	
5,649,176 A		7/1997	Selvidge et al.	716/16
5,659,716 A		8/1997	Selvidge et al.	703/23
5,701,441 A	*	12/1997	Trimberger	326/39
5,761,484 A		6/1998	Agarwal et al.	
5,777,489 A		7/1998	Barbier et al.	
5,847,578 A		12/1998	Noakes et al.	
5,854,752 A		12/1998	Agarwal	
5,920,712 A		7/1999	Kuijsten	
5,960,191 A	*	9/1999	Sample et al.	703/23

FOREIGN PATENT DOCUMENTS

WO	WO 94/06210	3/1994
WO	WO 94/23389	10/1994

OTHER PUBLICATIONS

Shibata, Y.; Miyazaki, H.; Xiao–Ping Ling; Amano, H., Towards the realistic "virtual hardware," Innovative Architecture for Future Generation High–Performance Processors and Systems, 1997, 1998 pp.: 50–55.*

Varghese, J.; Butts, M.; Batcheller, J., "An efficient logic emulation system," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol.: 1 Issue: 2, Jun. 1993 pp.: 171–174.*

Lo, W.Y.; Choy, C.S.; Chan, C.F., "Hardware emulation board based on FPGAs and Programmable Interconnections," Proceedings of the Fifth International Workshop on Rapid System Prototyping, 1994. Shortening the Path from Spec to Prototype. pp.: 126–130.*

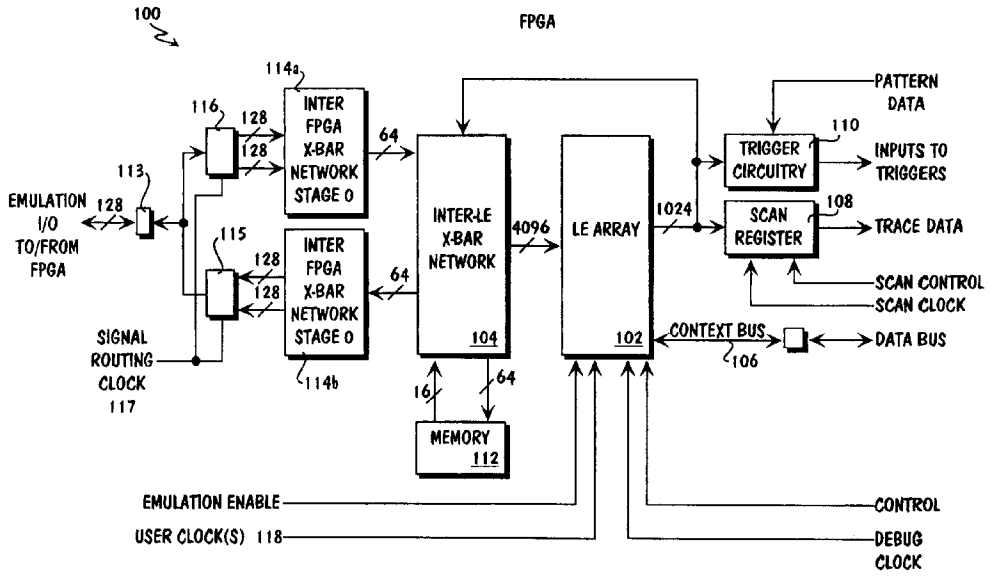
(Continued)

Primary Examiner—B. D. Thomson
(74) *Attorney, Agent, or Firm*—Banner & Witcoff, Ltd.

(57) **ABSTRACT**

A regionally time multiplexed emulation system includes an emulator for emulating a circuit design. The emulator includes a plurality of reconfigurable logic devices with buffered I/O pins and reconfigurable logic elements. The reconfigurable logic devices are reconfigurable to emulate a circuit design using at least one user clock to clock the logic elements and at least one signal routing clock to time multiplex the routing of emulation signals between the reconfigurable logic devices, with the at least one signal routing clock being independent of the at least one user clock.

20 Claims, 11 Drawing Sheets



OTHER PUBLICATIONS

Babb, et al., "Logic Emulation with Virtual Wires," IEEE Transactions on Computer–Aided Design of Integrated Circuits and Systems, vol. 16, No. 6, Jun. 1997, pp. 609–626.

Berger, "Teramac HW Simulator System External Reference Specification," Nov. 7, 1991, Revision 1.1, pp. 3–37.

Snider et al., "The PLASMA Chip Specification,"Aug. 1, 1995, pp. 1–124.

Snider et al., "The Teramac Configurable Compute Engine," Field Programmable Logic and Applications, 5th International Workshop, FPL '95 Oxford, United Kingdom, pp. 44–53.

XILINX, "The Programmable Gate Array Design Handbook," First Edition, 1986, pp. i–A–10.

"Translation of an Office Action of Japanese Patent Office," from a Japanese counterpart application, 7 pgs., Jul. 2, 2002.

Office Action of Japanese Patent Office, from the Japanese counterpart application, 6 pgs., Jul. 2, 2002.

* cited by examiner

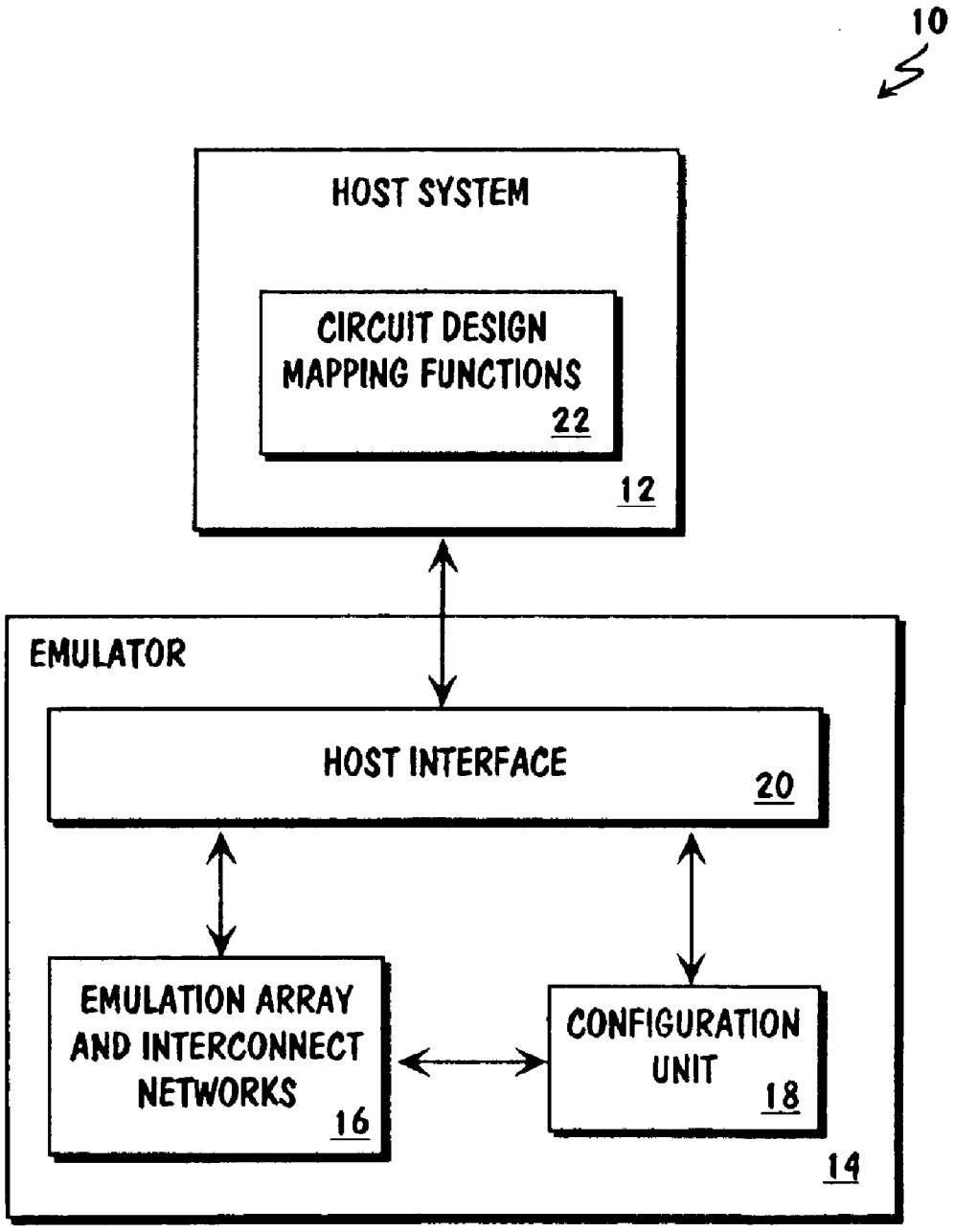

10
HOST SYSTEM
CIRCUIT DESIGN MAPPING FUNCTIONS
22
12
EMULATOR
HOST INTERFACE
20
EMULATION ARRAY AND INTERCONNECT NETWORKS
16
CONFIGURATION UNIT
18
14

FIG. 1

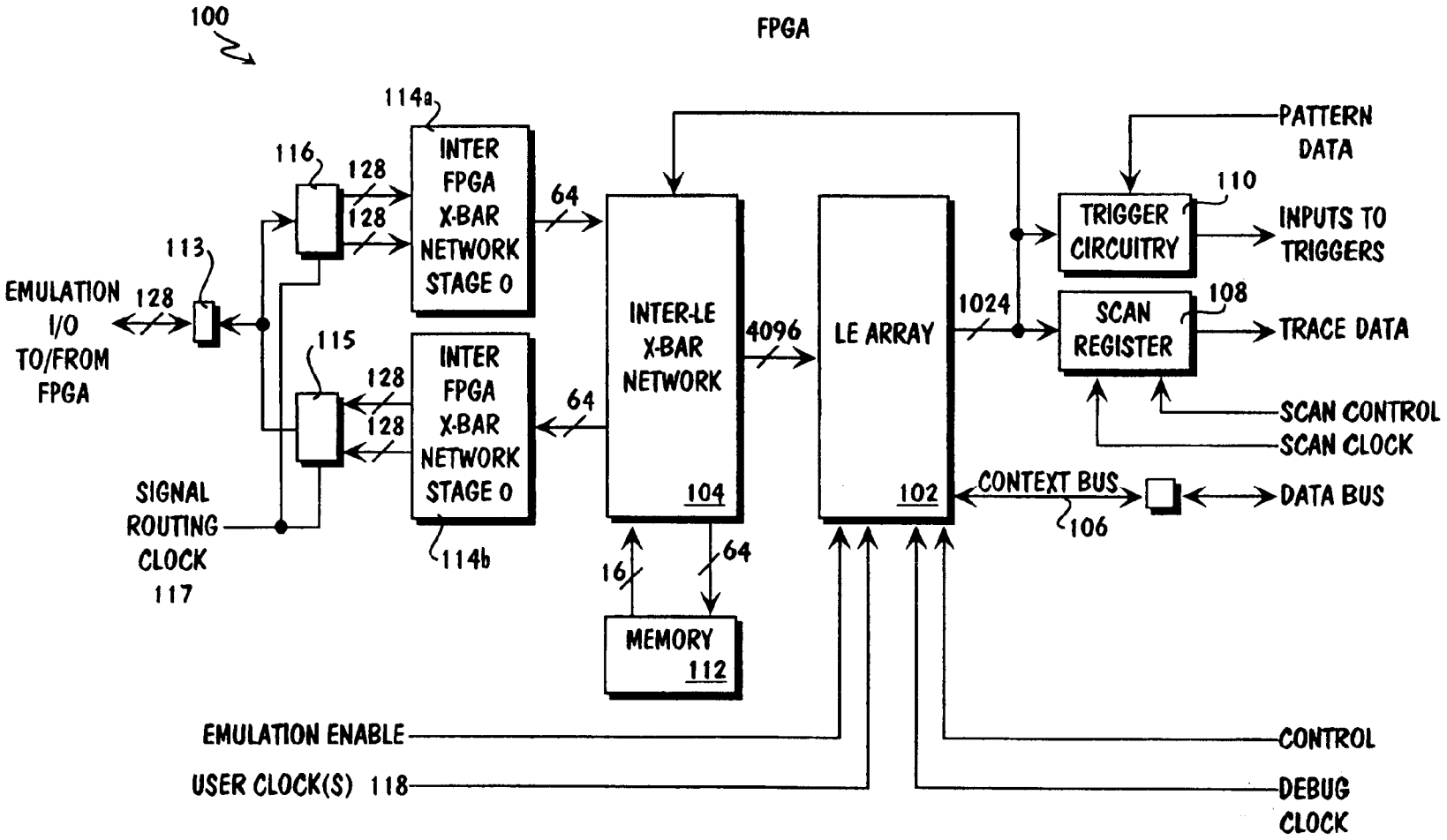

100
FPGA
114a
INTER FPGA X-BAR NETWORK STAGE 0
116
128
128
64
113
EMULATION I/O TO/FROM FPGA
128
115
INTER FPGA X-BAR NETWORK STAGE 0
128
128
64
114b
SIGNAL ROUTING CLOCK 117
INTER-LE X-BAR NETWORK
104
4096
LE ARRAY
102
1024
PATTERN DATA
TRIGGER CIRCUITRY
110
INPUTS TO TRIGGERS
SCAN REGISTER
108
TRACE DATA
SCAN CONTROL
SCAN CLOCK
CONTEXT BUS
106
DATA BUS
16
64
MEMORY 112
EMULATION ENABLE
USER CLOCK(S) 118
CONTROL
DEBUG CLOCK

FIG. 2

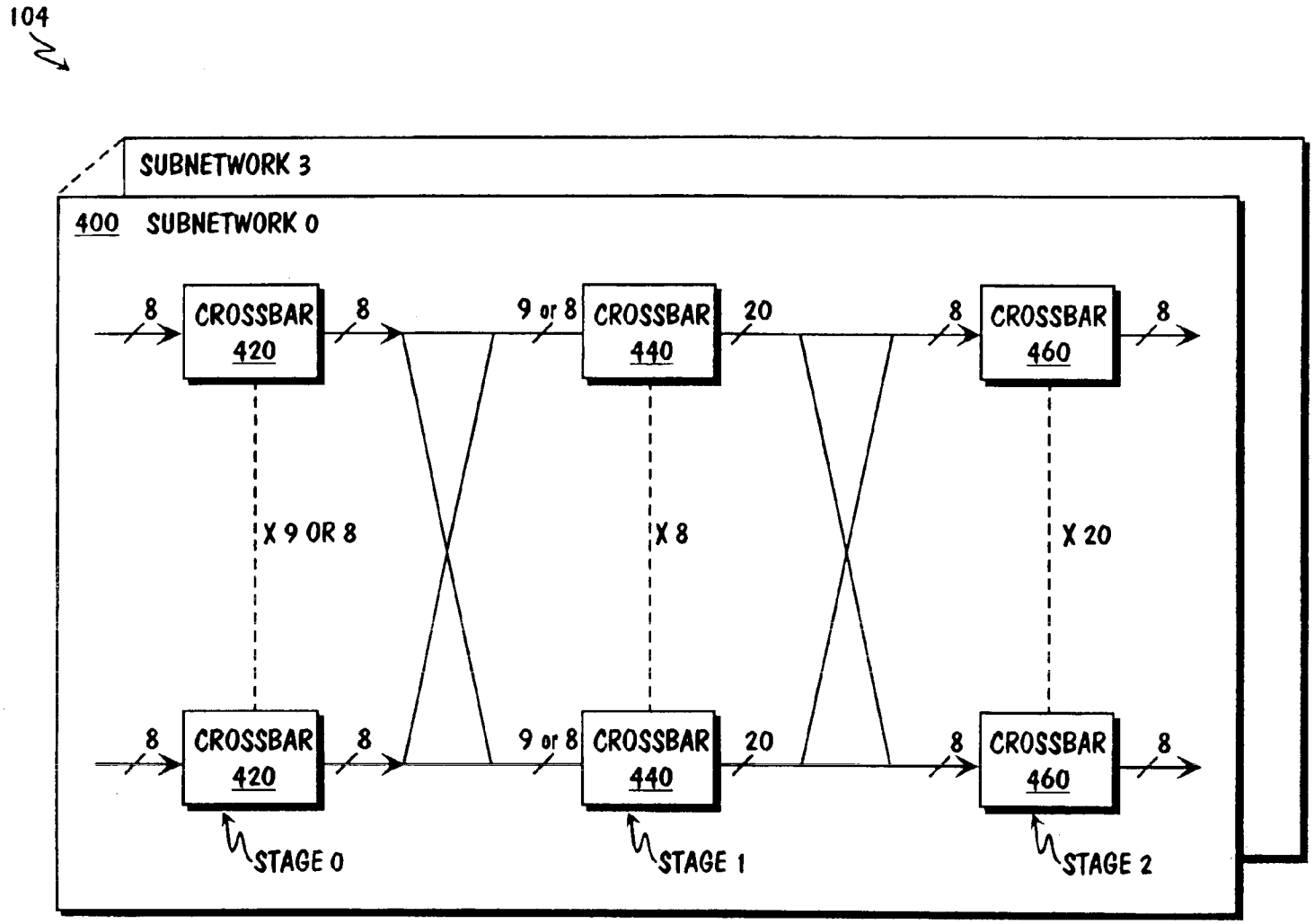

104
SUBNETWORK 3
400 SUBNETWORK 0
8
CROSSBAR 420
8
9 or 8
CROSSBAR 440
20
8
CROSSBAR 460
8
X 9 OR 8
X 8
X 20
8
CROSSBAR 420
8
9 or 8
CROSSBAR 440
20
8
CROSSBAR 460
8
STAGE 0
STAGE 1
STAGE 2

FIG. 3

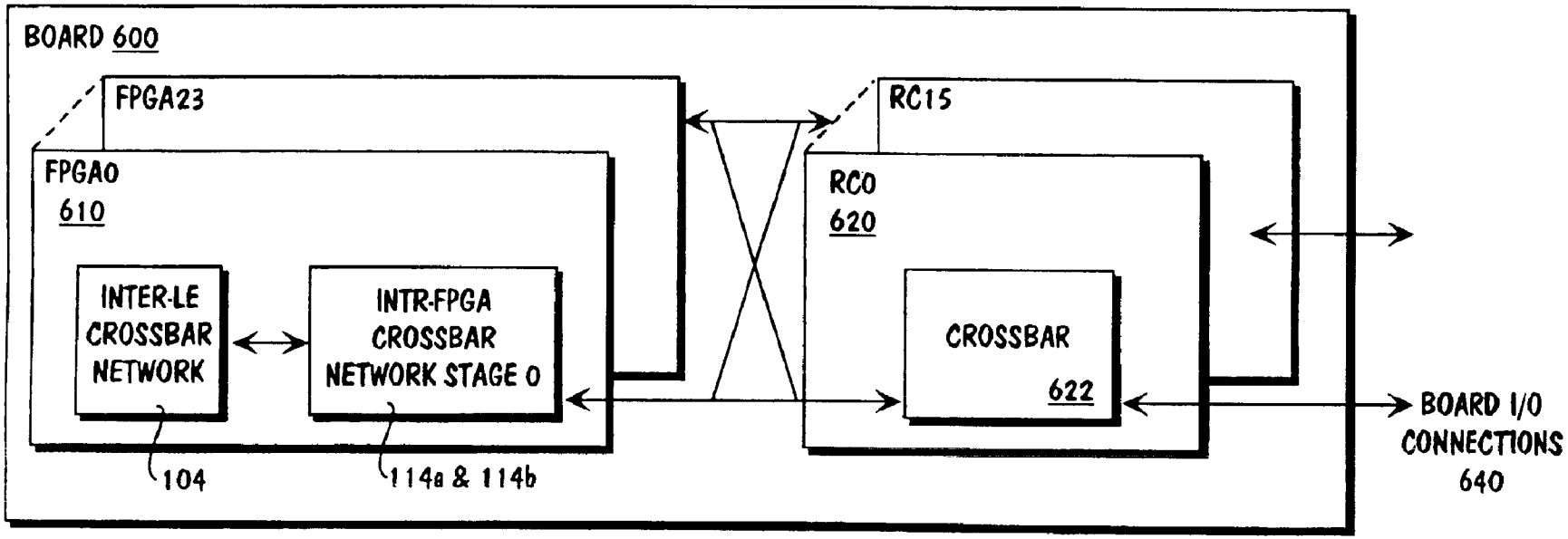

BOARD 600
FPGA23
FPGA0
610
INTER-LE
CROSSBAR
NETWORK
INTR-FPGA
CROSSBAR
NETWORK STAGE 0
104
114a & 114b
RC15
RC0
620
CROSSBAR
622
BOARD I/O
CONNECTIONS
640

FIG. 4

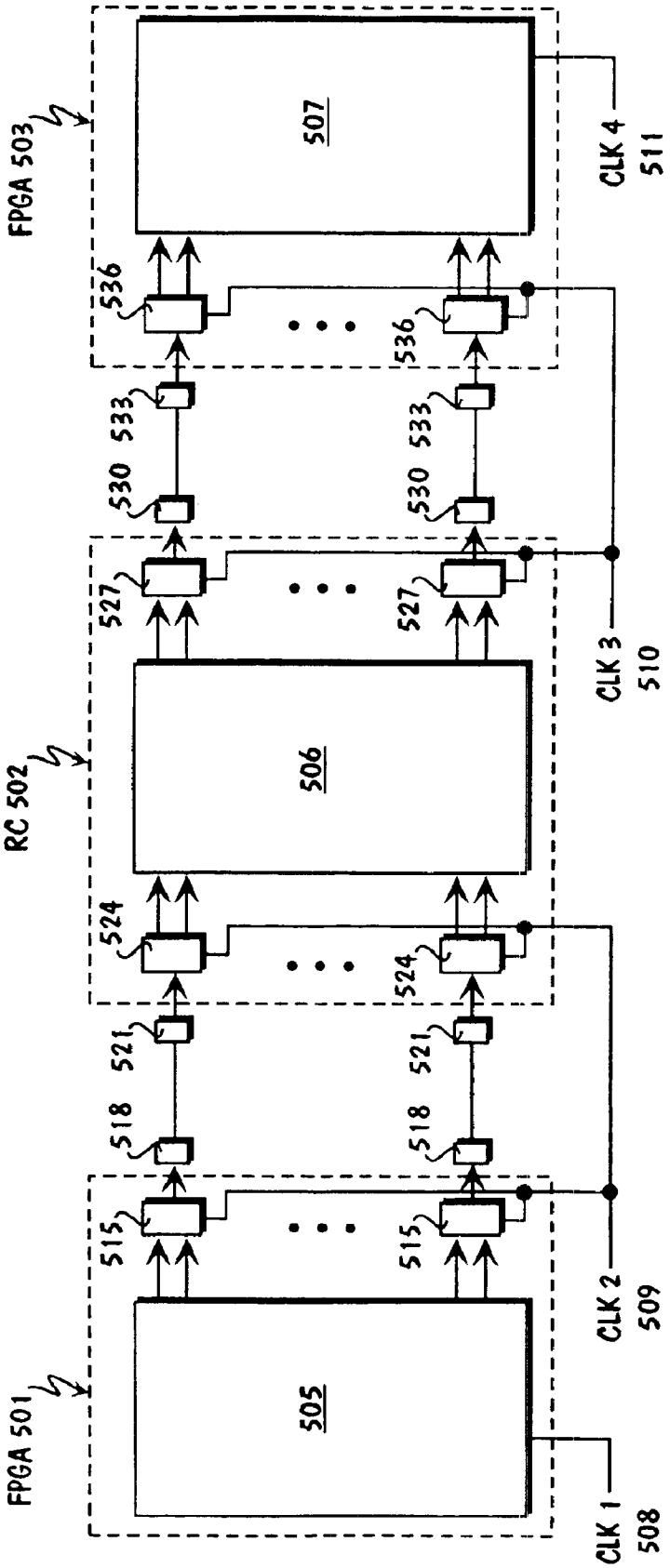

FPGA 503
507
CLK 4
511
536
536
536
533
533
530
530
RC 502
506
CLK 3
510
527
527
527
524
524
524
521
521
518
518
FPGA 501
505
CLK 2
509
CLK 1
508
515
515
515

FIG. 5

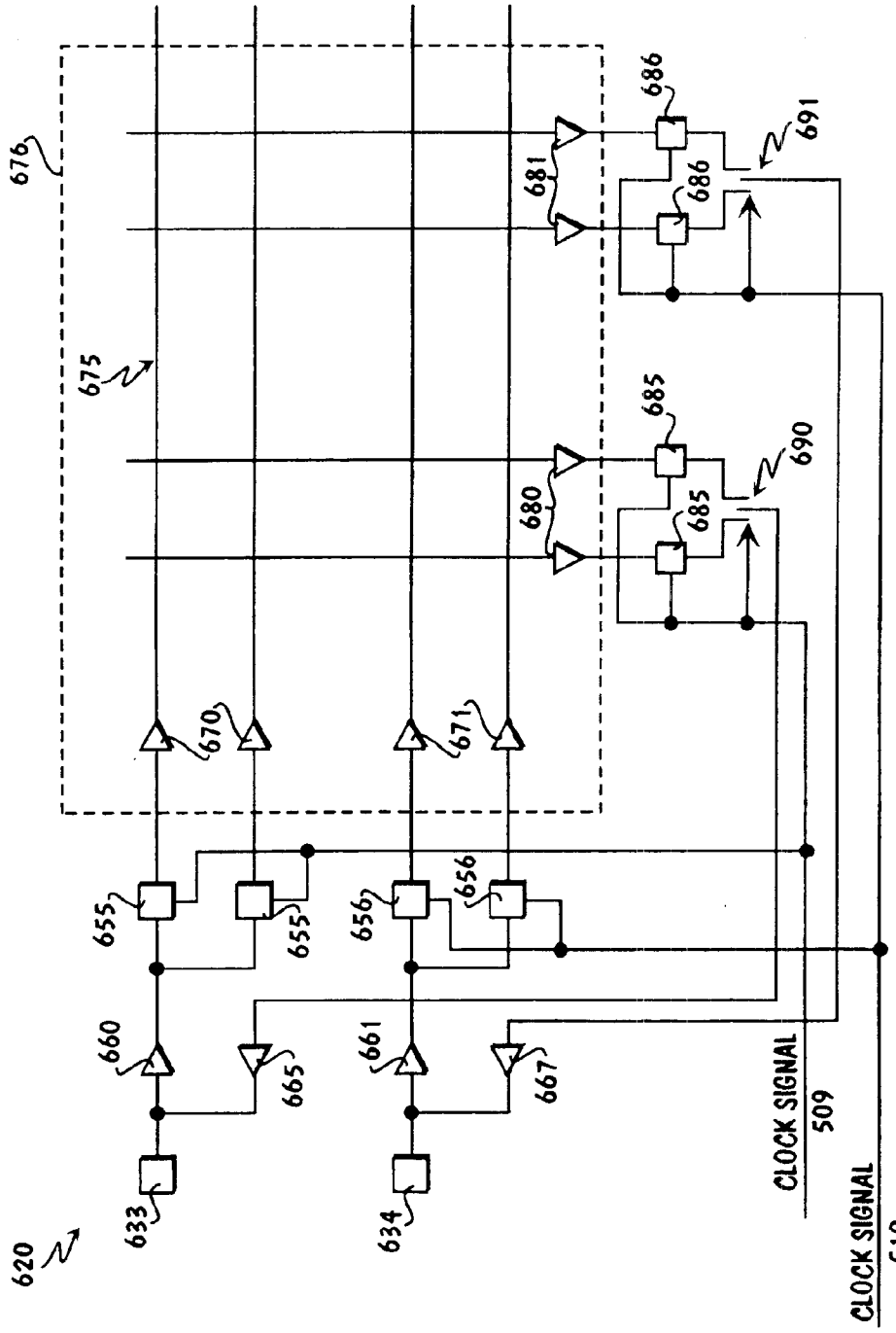

676
675
686
691
681
686
685
690
680
685
670
671
655
655
656
656
660
665
661
667
633
634
620
CLOCK SIGNAL
509
CLOCK SIGNAL
510

FIG. 6

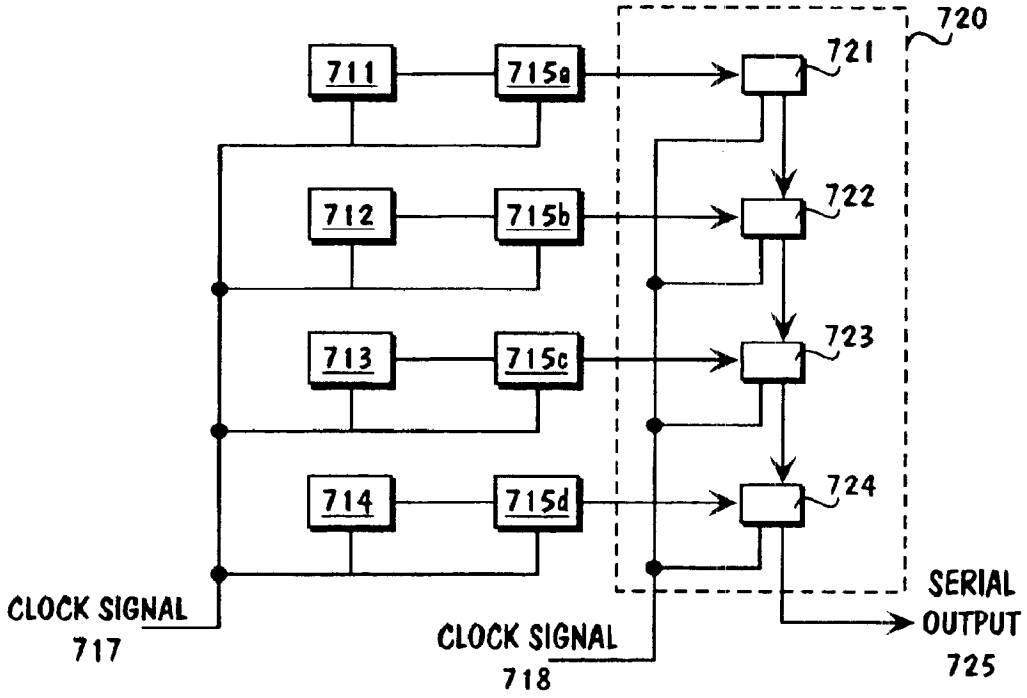

720
711
715a
721
712
715b
722
713
715c
723
714
715d
724
CLOCK SIGNAL
717
CLOCK SIGNAL
718
SERIAL
OUTPUT
725

FIG. 7a

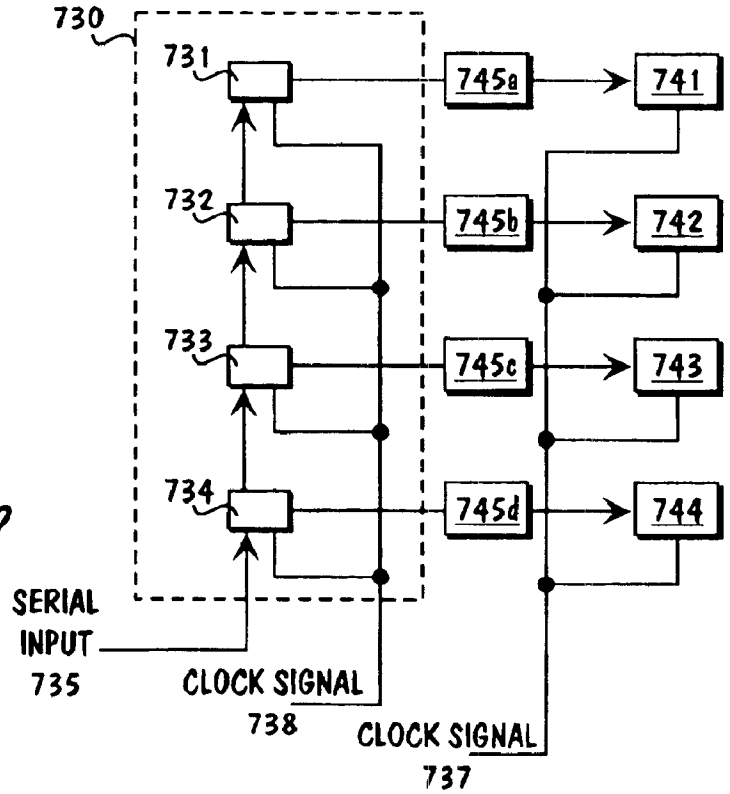

730
731
745a
741
732
745b
742
733
745c
743
734
745d
744
SERIAL
INPUT
735
CLOCK SIGNAL
738
CLOCK SIGNAL
737

FIG. 7b

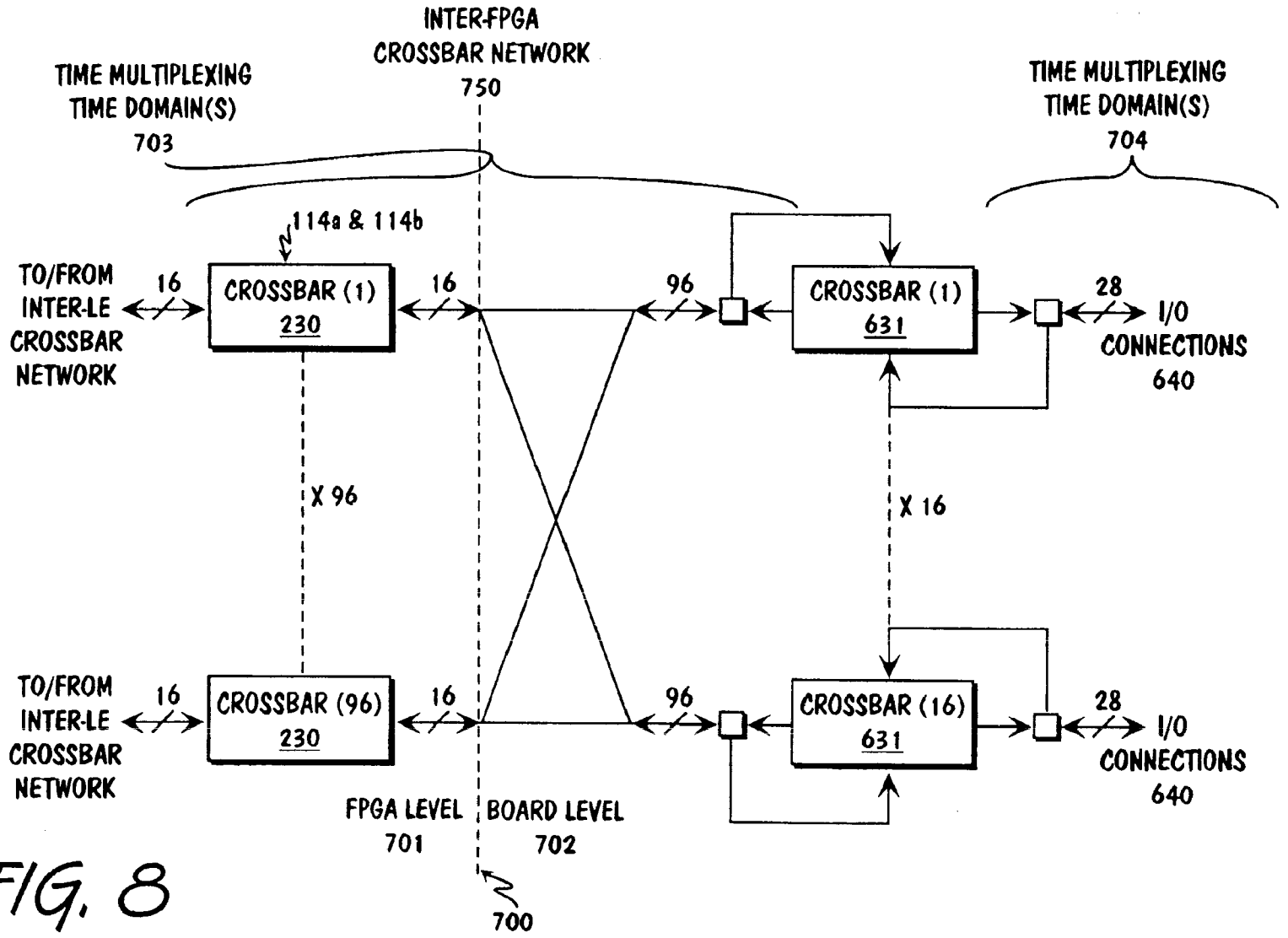

INTER-FPGA
CROSSBAR NETWORK
750
TIME MULTIPLEXING
TIME DOMAIN(S)
703
TIME MULTIPLEXING
TIME DOMAIN(S)
704
114a & 114b
TO/FROM
INTER-LE
CROSSBAR
NETWORK
16
CROSSBAR (1)
230
16
96
CROSSBAR (1)
631
28
I/O
CONNECTIONS
640
X 96
X 16
TO/FROM
INTER-LE
CROSSBAR
NETWORK
16
CROSSBAR (96)
230
16
96
CROSSBAR (16)
631
28
I/O
CONNECTIONS
640
FPGA LEVEL
701
BOARD LEVEL
702
700

FIG. 8

A561

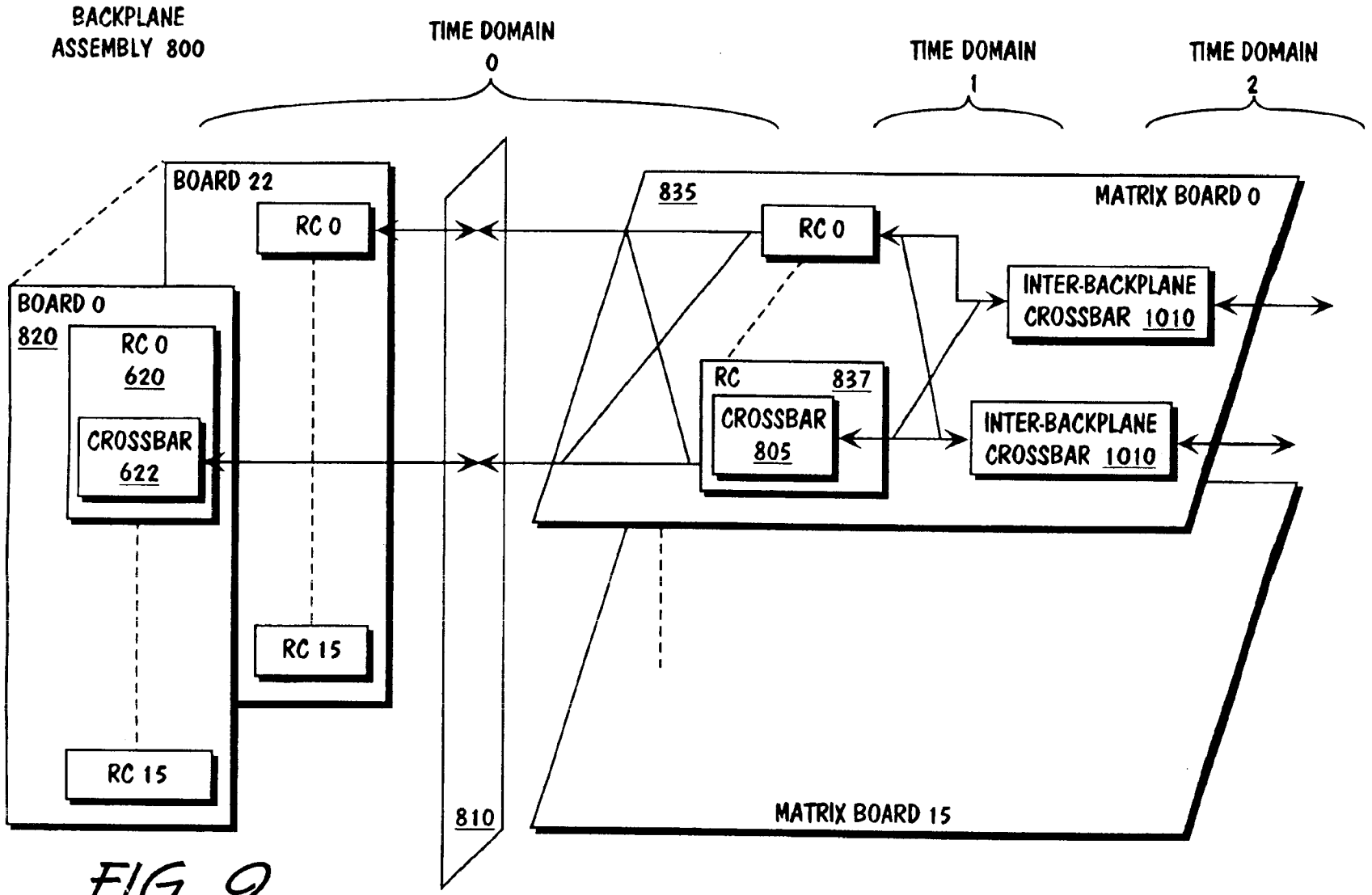

BACKPLANE ASSEMBLY 800
TIME DOMAIN 0
TIME DOMAIN 1
TIME DOMAIN 2
BOARD 22
RC 0
RC 15
BOARD 0
820
RC 0
620
CROSSBAR
622
RC 15
810
835
MATRIX BOARD 0
RC 0
INTER-BACKPLANE CROSSBAR 1010
RC 837
CROSSBAR
805
INTER-BACKPLANE CROSSBAR 1010
MATRIX BOARD 15

FIG. 9

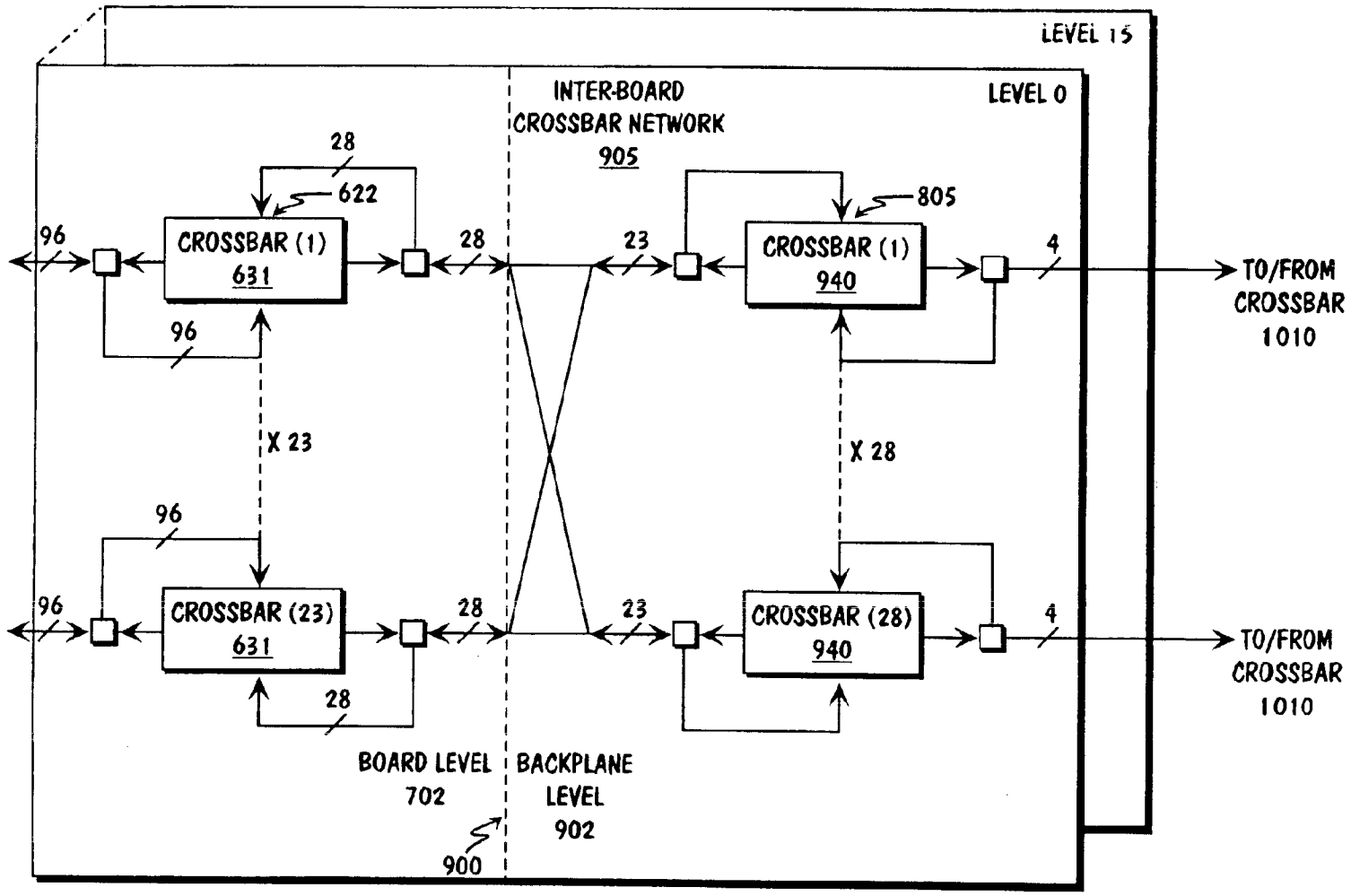

LEVEL 15
INTER-BOARD
CROSSBAR NETWORK
905
LEVEL 0
28
622
805
96
CROSSBAR (1)
631
28
23
CROSSBAR (1)
940
4
TO/FROM
CROSSBAR
1010
96
X 23
X 28
96
96
CROSSBAR (23)
631
28
23
CROSSBAR (28)
940
4
TO/FROM
CROSSBAR
1010
28
BOARD LEVEL
702
BACKPLANE
LEVEL
902
900

FIG. 10

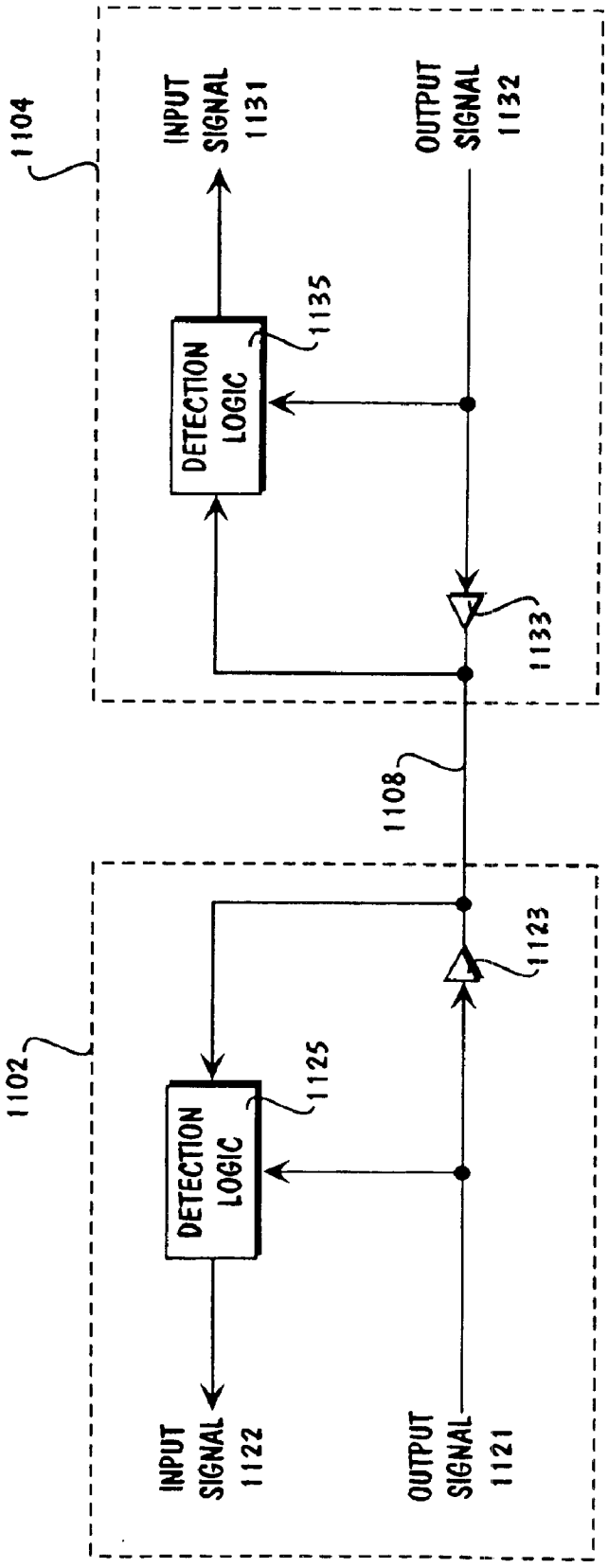

1104
INPUT SIGNAL 1131
OUTPUT SIGNAL 1132
1135
DETECTION LOGIC
1133
1108
1123
1102
1125
DETECTION LOGIC
INPUT SIGNAL 1122
OUTPUT SIGNAL 1121

FIG. 11

REGIONALLY TIME MULTIPLEXED EMULATION SYSTEM

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to the field of emulation systems. More specifically, the present invention relates to methodology for increasing capacity of an emulation system.

2. Background Information

Emulation systems for emulating circuit designs are known in the art. Typically, prior art emulation systems are formed using conventional general purpose field programmable gate arrays (FPGAs) and general purpose routing chips. A circuit design to be emulated is "realized" on the emulation system by compiling a "formal" description of the circuit design, and mapping the circuit design onto the logic elements (LEs) of the FPGAs and the routing chips.

As circuit designs have become larger and larger, up to and including designs having millions of transistors, a similar increase in size of emulation systems has become necessary in order to emulate such circuit designs. Large emulation systems typically include a significant number of FPGAs as well as a significant number of routing chips to route signals between the FPGAs. However, given the large number of FPGAs which may be included in an emulation system, the number of routing chips required to provide adequate flexibility to concurrently route large numbers of input and output signals to and from an FPGA has become prohibitively expensive.

An article by Jonathan Babb et al. entitled "Logic Emulation with Virtual Wires". (hereinafter "Babb et al.") provides one solution to this problem, referred to as "time multiplexing" or the use of "virtual wires". Using time multiplexing, multiple logical outputs of an FPGA share a single physical output with only one of the logical outputs being able to output a signal on the single physical output in any given clock cycle. Thus, the logical outputs are multiplexed on the single physical output over time. Similarly, a physical input to an FPGA is shared by multiple logical inputs with only one of the logical inputs being able to receive an input signal on the physical input in any given clock cycle. All of the FPGAs in the Babb et al. system, as well as any routing chips interconnecting the FPGAs, are clocked by the same clock signal (see, Babb et al., p. 5, § 2.1).

One problem with the Babb et al. system is that it is primarily designed to emulate synchronous logic providing synchronous signals, and does not support time multiplexing of asynchronous signals for emulating asynchronous logic. Rather, such asynchronous signals must be hard-wired to dedicated FPGA physical inputs and outputs, while the interconnection of time multiplexed synchronous signals is automatically configured for the user (see, Babb et al., p. 5, § 2.1).

Additionally, even with the use of time multiplexing, or in systems where asynchronous signals are hard-wired to dedicated inputs and outputs, other problems still exist. One such problem is that of synchronizing clock signals in the emulation system. Despite the use of time multiplexing to reduce overall system size, the system can still remain relatively large. Such systems can range in size up to a few meters square. Synchronizing high frequency clock signals across such a large area creates a significant problem.

Thus, it is desirable to have an emulation system with improved capacity without the disadvantages of conventional time multiplexing. As will be described in more detail below, the present invention provides for an emulation system that achieves these and other desired results, which will be apparent to those skilled in the art from the description to follow.

SUMMARY OF THE INVENTION

A regionally time multiplexed emulation system is described herein. The emulator includes a plurality of reconfigurable logic devices with buffered I/O pins and reconfigurable logic elements. The reconfigurable logic devices are reconfigurable to emulate a circuit design using at least one user clock to clock the logic elements and at least one signal routing clock to time multiplex the routing of emulation signals between the reconfigurable logic devices, with the at least one signal routing clock being independent of the at least one user clock. As a result, both asynchronous as well as synchronous signals may be automatically routed by the mapping software of the emulation system.

BRIEF DESCRIPTION OF DRAWINGS

The present invention will be described by way of exemplary embodiments, but not limitations, illustrated in the accompanying drawings in which like references denote similar elements, and in which:

FIG. **1** is a block diagram showing an exemplary emulation system which incorporates the teachings of the present invention;

FIG. **2** is a block diagram showing an exemplary reconfigurable logic device which may be used with one embodiment of the present invention;

FIG. **3** is a block diagram showing an inter-LE crossbar network according to one embodiment of the present invention;

FIG. **4** is a block diagram of a circuit board which can be used in an emulator according to one embodiment of the present invention;

FIG. **5** is a block diagram illustrating the concept of regional time multiplexing according to one embodiment of the present invention;

FIG. **6** is a block diagram showing one embodiment of a multi-clocked routing chip suitable for use with one embodiment of the present invention;

FIGS. **7***a* and **7***b* are block diagrams illustrating shift registers which may be used to support the regional time multiplexing according to one embodiment of the present invention.

FIG. **8** is a block diagram showing a logical view of an inter-reconfigurable logic device crossbar network according to one embodiment of the present invention;

FIG. **9** is a block diagram of a backplane assembly according to one embodiment of the present invention;

FIG. **10** is a block diagram illustrating a logical view of an inter-board crossbar network according to one embodiment of the present invention; and

FIG. **11** is a block diagram illustrating the concurrent bi-directional data transfer over a single connection according to one embodiment of the present invention.

DETAILED DESCRIPTION

In the following description, for purposes of explanation, specific numbers, materials and configurations are set forth

in order to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that the present invention may be practiced without the specific details. In other instances, well known features are omitted or simplified in order not to obscure the present invention.

FIG. **1** is a block diagram showing an exemplary emulation system which incorporates the teachings of the present invention. As illustrated, an emulation system **10** includes host system **12** and emulator **14**. Host system **12** includes in particular circuit design mapping functions **22** incorporated with the teachings of the present invention. In one embodiment, circuit design mapping functions **22** are implemented in software. In this embodiment, circuit design mapping software **22** is stored in a suitable storage medium (not shown) of host system **12**, and is loaded into memory (not shown) of host system **12** for execution by a processor (not shown) of host system **12**. Except for circuit design mapping functions **22**, host system **12** is intended to represent a broad category of host systems found in conventional emulation systems known in the art, and thus will not be otherwise discussed further.

Emulator **14** includes emulation array and interconnect networks **16** incorporated with the teachings of the present invention, a configuration unit **18** and host interface **20** coupled to each other as shown. Except for emulation array and interconnecting network **16**, emulator **14** is intended to represent a broad category of elements found in conventional emulators, whose functions and constitutions are well known to those skilled in the art, and therefore will not be otherwise further described either. As will be described in more detail below, emulation array and interconnect networks **16** comprises a number of reconfigurable logic elements (LEs) distributively packaged in a number of reconfigurable circuits and interconnected in a regional time multiplexing manner.

A particular example of an emulation array and interconnect networks **16** (suitable for incorporating the present invention) is disclosed in U.S. Pat. No. 5,574,388 to Barbier et al., which is hereby fully incorporated by reference. The manner in which regional time multiplexing is incorporated into emulation array and interconnect networks **16** will be described in more detail below.

FIG. **2** is a block diagram showing one embodiment of a reconfigurable logic device which may be used with one embodiment of the present invention. The embodiment is of a custom or special purpose field programmable gate array (FPGA) type, hereinafter simply FPGA. For the purpose of this application, the term "FPGA" is to mean all reconfigurable circuits, and not just the typical general purpose FPGAs available in the market. FPGA **100** includes LE array **102**, and buffered I/O pins **113**. LE array **102** includes multiple reconfigurable LEs clocked by user clock(s) **118**. As is well known in the art, the reconfigurable LEs are used to "realize" various logic elements of circuit designs, whereas, buffered I/O pins **113** are used to provide time multiplexed inputs/outputs to/from FPGA **100**. Each of buffered I/O pins **113** can be statically configured to be either an input or an output pin. This static configuration can be accomplished in any of a wide variety of conventional manners, such as by way of a configuration register.

More importantly, as illustrated in FIG. **2**, each of buffered I/O pins **113** is an input/output for multiple different logical inputs/outputs. In the illustrated embodiment, for ease of explanation, each buffered I/O pin **113** is an input/output for two different logical inputs/outputs, however, in alternate embodiments each buffered I/O pin **113** is an input/output for three or more different logical inputs/outputs. The logical inputs/outputs correspond to inputs to/outputs from inter-FPGA crossbar network stage 0 **114***a*/**114***b*. These logical inputs/outputs are time multiplexed on buffered I/O pins **113** by I/O circuitry **115**, which includes a two-to-one multiplexer, and I/O circuitry **116**, which includes a one-to-two demultiplexer, using signal routing clock **117**. As illustrated, only 32 buffered I/O pins **113** are necessary to support the 64 logical inputs/outputs due to the two-to-one multiplexing performed by I/O circuitry **115** and **116**.

As illustrated in FIG. **2**, I/O circuitry **115** and **116** are clocked by signal routing clocks **117** whereas the LEs are clocked by a different clock signal (or signals), user clock(s) **118**. Except for the relationship that each of signal routing clock **117** having a higher frequency than an associated user clock **118**, signal routing clocks **117** are independent of user clocks **118**. For the purpose of this application the "associated" user clock of a signal routing clock is the user clock employed to clock the logic elements from which the I/O signals of the I/O pins clocked by the signal routing clock originate or destined for.

In the illustrated embodiment of FIG. **2**, each signal routing clock **117** is of a higher frequency than the "associated" user clock **118**, thereby allowing signals to be output from FPGA **100** more frequently than they are changed internally in FPGA **100**. Thus, signals can be advantageously transferred into and out of FPGA **100** asynchronously to the changing of the signals internal to FPGA **100**. Typically, the frequency of the clock signal(s) in the signal routing time domain is 10 to 100 times greater than the frequency of the clock signal(s) in the user time domain. However, different embodiments may have different frequency ratios.

One embodiment of I/O circuitry **115** and **116** of each of the buffered I/O pins **113** is clocked by the same signal routing clock **117**. In alternate embodiments, I/O circuitry **115** and **116** for different buffered I/O pins **113** can be clocked by different signal routing clocks rather than a single signal routing clock.

Preferably, FPGA **100** also includes memory **112**, context bus **106**, scan register **108**, and trigger circuitry **110**. Memory **112** facilitates usage of FPGA **100** to emulate circuit design with memory elements. Context bus **106**, scan register **108** and trigger circuitry **110** provide on-chip integrated debugging facility for FPGA **100**. These elements are described in U.S. patent application Ser. No. 08/542,838, entitled "A Field Programmable Gate Array with Integrated Debugging Facilities", which is hereby fully incorporated by reference.

Inter-LE crossbar network **104** is also integrated into FPGA **100**. Inter-LE crossbar network **104** interconnects the LEs of LE array **102**, memory **112**, and buffered I/O pins **113** of FPGA **100**, to be described more fully below.

Additionally, according to one embodiment, a corresponding portion of inter-FPGA crossbar network stage 0 **114***a*/**114***b* is also advantageously integrated into FPGA **100**. The various portions of inter-FPGA crossbar network stage 0 **114***a*/**114***b* together with the remainder of inter-FPGA crossbar network interconnect FPGAs **100** of a logic board and the I/O connections of the logic board, which will also be described in more detail below.

In one embodiment, LE array **102** includes **128** reconfigurable LEs, while memory **112** uses 8-bit input and 8-bit output, and FPGA **100** has 32 buffered I/O pins **113**.

FIG. **3** is a block diagram showing an inter-LE crossbar network according to one embodiment of the present invention. For the illustrated embodiment, inter-LE crossbar network **104** includes four subnetworks of crossbars **400**. A crossbar device is an interconnect device which receives multiple inputs and maps the inputs to multiple outputs of the device. Each input can be mapped to any of the multiple outputs. Which inputs are mapped to which outputs are identified by programming the crossbar device. Such crossbar devices arc well known to those skilled in the art and thus will not be described further except as they pertain to the present invention.

For the illustrated embodiment, the first two subnetworks **400** are used to map **72** inputs to 160 outputs, whereas the second two subnetwork **400** are used to map 64 inputs to 160 outputs. Each subnetwork **400** comprises three stages, stage 0, stage 1, and stage 2. Stage 0 of the first two subnetworks **400** include nine 8×8 crossbars **420**, whereas stage 0 of the last two of subnetworks **400** include eight 8×8 crossbars **420**. In turn, stage 1 of the first two subnetworks **400** include eight 9×20 crossbars **440**, whereas stage 1 of the last two subnetworks **400** include eight 8×20 crossbars **440**. Stage 2 of all four subnetworks **400** include twenty 8×8 crossbars **460**.

Having now described the FPGAs including the manner in which their LEs are interconnected on-chip and to the FPGA I/O pins, we now proceed to describe how the FPGAs are interconnected together on a logic board and to the logic board's I/O pins.

FIG. **4** is a block diagram of a circuit board which can be used in an emulator according to one embodiment of the present invention. A circuit board **600** is shown comprising multiple FPGAs **610** and multi-clocked routing chips (RCs) **620** coupled to each other in a "butterfly" manner as shown. In one implementation, each of the FPGAs **610** is an FPGA **100** of FIG. **2**. Each multi-clocked RC **620** includes a crossbar **622** and related circuitry for supporting regional time multiplexing.

Recall from the earlier description that inter-FPGA network stage 0 is distributively implemented on FPGAs **610**. Collectively, RCs **620** implement inter-FPGA network stage 1. Together, the two stages interconnect FPGAs **610** on circuit board **600** and to the I/O pins **640** of circuit board **600**. (As will be discussed in more detail below, inter-FPGA network stage 1 also "doubles up" as inter-board network stage 0.)

Thus, signals output by any of the FPGAs **610** can be routed to any other FPGA **610** on circuit board **600** or routed off-board, either case, through multi-clocked RCs **620**. Similarly, input signals to circuit board **600** can be routed to any one of the on-board FPGAs **610** or rerouted off-board. Each of the multi-clocked RCs **620** can advantageously operate in multiple different signal routing time domains, with one set of at least one I/O pin being clocked according to one signal routing time domain while another set of at least one I/O pin is clocked according to another signal routing time domain. Thus, the signals which are transferred into and out of multi-clocked RCs **620** are time multiplexed and different time domains can be distributed throughout different regions of the emulator. This regional time multiplexing is discussed in more detail below.

In the embodiment shown, board **600** includes twenty-four FPGAs **610** and sixteen RCs **620**. However, it is to be appreciated that alternate embodiments can include different numbers of FPGAs and RCs.

FIG. **5** is a block diagram illustrating the concept of regional time multiplexing according to one embodiment of the present invention. Two FPGAs **501** and **503** and a multi-clocked RC **502** are illustrated. In the illustrated embodiment, FPGAs **501** and **503** are both FPGAs **100** of FIG. **2**, and can be situated on the same or different boards **600** of FIG. **4**. It is to be appreciated that, depending on their locations within the emulator, additional multi-clocked RCs **502** may be needed to route signals between FPGAs **501** and **503**. For ease of illustration, the internal circuitry of FPGAs **501** and **503** are shown as blocks **505** and **507**, and are intended to represent the reconfigurable logic elements, inter-logic element crossbar network and inter-FPGA crossbar network stage 0, as well as other internal circuitry, of the FPGAs as illustrated in FIG. **2**.

As illustrated, the internal circuitry of FPGA **501** is clocked in a user time domain by clock signal **508** (clk**1**), whereas the I/O circuitry **515** for the input/output of signals is clocked in a signal routing time domain by clock signal **509** (clk**2**). As discussed above, except for clock signal **509** (clk**2**) being of a higher frequency than clock signal **508**, clock signals **508** and **509** (clk**2**) are independent of one another.

The output signals from the internal circuitry **505** of FPGA **501** are input to two-to-one multiplexers of I/O circuitry **515** and output from FPGA **501** via pins **518**. Input signals to RC **502** are received on I/O pins **521** and provided to I/O circuitry **524** where the signals are demultiplexed and input to static routing circuitry **506** of RC **502**. The outputs of static routing circuitry **506** are provided to I/O circuitry **527**. Each of I/O circuitry **527** also includes a two-to-one multiplexer, providing an output signal to one of the I/O pins **530**. The output signals are routed to I/O pins **533** of FPGA **503**, and then demultiplexed by demultiplexers of I/O circuitry **536** and input to internal circuitry **507** of FPGA **503**.

In the illustrated embodiment, static routing circuitry **506** of FIG. **5** is a crossbar **622** of FIG. **4**. Static routing circuitry **506** is configured to route particular inputs to particular outputs as part of the programming process of the emulator. Given the static nature of circuitry **506**, the circuitry **506** is not clocked.

As illustrated in FIG. **5**, internal circuitry **505** of FPGA **501** is clocked in a user time domain by clock signal **508** (clk**1**), multiplexers **515** and demultiplexers **524** are clocked in a signal routing time domain by clock signal **509** (clk**2**), multiplexers **527** and demultiplexers **536** are clocked in another signal routing time domain by clock signal **510** (clk**3**), and internal circuitry **507** of FPGA **503** is clocked in another user time domain by clock signal **511** (clk**4**).

Thus, two user time domains and two signal routing time domains are illustrated in FIG. **5**, as clocked by clock signals **508**, **509**, **510**, and **511**. As illustrated, different sets of I/O pins and related I/O circuitry of RC **502** are clocked by different clock signals. Thus, signals can be advantageously transferred out of RC **502** asynchronously to the input of signals to RC **502** by outputting the signals from a different set than the signals were input on. In an alternate embodiment of the present invention, user clock signals **508** and **511** are the same clock signal. Thus, in this alternate embodiment, internal circuitry **505** and **507** are both in the same time domain.

In an alternate embodiment of the present invention, signals are routed directly from I/O pins **518** of FPGA **501** to/from I/O pins **533** of FPGA **503** without being routed through RC **502**. I/O circuitry **515** and **536** are both clocked by one of either signal routing clock **509** or signal routing clock **510**. Thus, even though a routing chip is not used in this alternate embodiment, the signal routing between

FPGAs is still clocked by a signal independent of the user clock signal(s).

Thus, FIG. **5** illustrates regional time multiplexing in which two different signal routing time domains are distributed throughout the emulator. As illustrated, different regions of the emulator are clocked according to different signal routing time domain clock signals. It is to be appreciated that additional signal routing time domains (not shown) can also be distributed throughout the emulator.

In an alternate embodiment of the present invention, signals are routed directly from I/O pins **510** of FPGA **501** to/from I/O pins **533** of FPGA **503** without being routed through RC **502**. I/O circuitry **515** and **536** are both clocked by one of either signal routing clock **509** or signal routing clock **510**. Thus, even though a routing chip is not used in this alternate embodiment, the signal routing between FPGAs is still clocked by a signal independent of the user clock signal(s).

In another alternate embodiment of the present invention, signal routing clock **509** and signal routing clock **510** are the same clock signal. Thus, although all inputs/outputs of the RC **502** are clocked by the same signal routing clock signal in this alternate embodiment, the signal routing clock **510** is still independent of the user clocks **508** and **511**. Thus, information can still be input to/output from FPGAs asynchronously to the changing of signals within the internal circuitry of the FPGAs.

FIG. **6** is a block diagram showing one embodiment of a multi-clocked RC **620** suitable for use in circuit board **600** in more detail. For ease of explanation, only two I/O pins **633** and **634** and associated circuitry are illustrated in FIG. **6**. It is to be appreciated that the remaining I/O pins of RC **620** have similar associated circuitry. I/O pin **633** is enabled as either an input or an output by driver **660** and driver **665**. Driver **660** is enabled if I/O pin **633** is to be an input, and driver **665** is enabled if I/O pin **633** is to be an output. When operating as an input, signals received on pin **633** are provided to latches **655**, which latch in the value on pin **633** on the falling edge of clock signal **509**. These latched signals will be input to the interconnect **675** by drivers **670**. The enablement of driver **660** or driver **665** is performed as part of the programming of the emulator.

Outputs from RC **620** via I/O pin **633** are controlled by latches **685** and switch **690**. Outputs from interconnect **675** are provided to latches **685** via drivers **680**. Latches **685** are clocked by clock signal **509** and latch in a value from their respective drivers **680** on the rising edge of clock signal **509**. The outputs of latches **685** are provided to switch **690**, which is also controlled by the rising edge of clock signal **509**. The latched value from one of the latches **685** is output by switch **690**, as controlled by clock signal **509**.

Similarly, I/O pin **634** is enabled as either an input or an output by driver **661** and driver **667**. Driver **661** is enabled if I/O pin **634** is to be an input, and driver **667** is enabled if I/O pin **634** is to be an output. When operating as an input, signals received on pin **634** are provided to latches **656**, which latch in the value on pin **634** on the falling edge of clock signal **510**. These latched signals will be input to the interconnect **675** by drivers **671**.

Outputs from RC **620** via I/O pin **634** are controlled by latches **686** and switch **691**. Outputs from interconnect **675** are provided to latches **686** via drivers **681**. Latches **686** are clocked by clock signal **510** and latch in a value from their respective drivers **681** on the rising edge of clock signal **510** The outputs of latches **686** are provided to switch **691**, which is also controlled by the rising edge of clock signal **510**. The latched value from one of the latches **686** is output by switch **691**, as controlled by clock signal **510**.

In the embodiment illustrated in FIG. **6**, driver **670**, driver **680** and interconnect **675** are referred to as the "static" part of RC **620**, denoted by dashed box **676**. The static part of RC **620** does not operate based on clock signals, so signals can be sampled out of the static part without regard for the clock frequency at which they were sampled in. Similarly, latches **655** and **685**, drivers **660** and **665**, and switch **690** are referred to as the "dynamic" part of RC **620**.

Thus, as illustrated in FIG. **6**, two different inputs/outputs of RC **620** are operating in two different signal routing time domains, clocked by two different clock signals. This separation advantageously allows time domains to be changed by simply routing through an RC **620**. In other words, a signal can be input to RC **620** via I/O pin **633** in the time domain clocked by clock signal **509**, and output from RC **620** via I/O pin **634** in the time domain clocked by clock signal **510**.

In an alternate embodiment of the present invention, the latches **685** and **686** are not included, and the output of drivers **680** and **681** are input directly to switches **690** and **691**, respectively. In this alternate embodiment, an additional latch (not shown), clocked by clock signal **509**, is situated between switch **690** and driver **665**, and another latch (not shown), clocked by clock signal **510**, is situated between switch **690** and driver **667**.

According to one embodiment of the present invention, RC **620** is clocked by two different signal routing clock signals, and the I/O pins are grouped in different sets, with each set being clocked in a different signal routing time domain. According to one implementation, the I/O pins on one side of RC **620** are part of a first set while the I/O pins on the other side of RC **620** are part of a second set.

According to alternate embodiments of the present invention, additional sets of I/O pins of RC **620** are clocked according to additional clock signals. A set of I/O pins can include a number of pins ranging from one to (x−1) where x is equal to the total number of I/O pins on RC **620**. Each of these different sets is clocked in a different time domain. An RC **620** can support up to x different signal routing time domains at any one time.

In the illustrated embodiment, multiplexers and demultiplexers are used to support the regional time multiplexing of the present invention. Alternate embodiments of the present invention can utilize any of a wide variety of conventional mechanisms for sharing of a single physical signal by multiple logical signals. FIGS. **7***a* and **7***b* illustrate one such alternate embodiment.

FIG. **7***a* is a block diagram illustrating an output register which may be used to support the regional time multiplexing according to one embodiment of the present invention. A parallel input, serial output shift register **720** is illustrated including four register cells **721**, **722**, **723**, and **724**. Inputs to register **720** are from internal circuitry **711**,**712**, **713**, and **714** through latches **715***a*–**715***d*. Internal circuitry **711**, **712**, **713**, and **714** can be any of a wide range of circuitry. Internal circuitry **711**, **712**,**713**, and **714** and latches **715***a*–**715***d* are clocked by internal clock signal **717**, and register **720** is clocked by time multiplexing clock signal **718**. Data is input to cells **721**, **722**, **723**, and **724** in parallel, then shifted out serially as serial output **725** starting with cell **724**. Thus, four logical internal signals, received from internal circuitry **711**, **712**, **713**, and **714**, are output via a single output signal **725**. In the illustrated embodiment, clock signal **718** has a frequency four times that of clock signal **717**. Thus, every clock signal **717** cycle a new set of four data signals can be

transferred to register **720**, with one signal being shifted out of register **720** every clock signal **718** cycle.

FIG. **7***b* is a block diagram illustrating an input register which may be used to support the regional time multiplexing according to one embodiment of the present invention. A serial input, parallel output shift register **730** is illustrated including four cells **731**, **732**, **733**, and **734**. Inputs to shift register **730** are shifted in serially from serial input **735**, with input data shifting from cell **734** up to cell **731**. In the illustrated embodiment, clock signal **738** has a frequency four times that of clock signal **737**. Thus, every clock signal **737** cycle a new set of four data signals can be transferred from register **730** to internal circuitry **741**, **742**, **743**, and **744** through latches **745***a*–**745***d*. Internal circuitry **741**, **742**, **743**, and **744** can be any of a wide range of circuitry.

FIG. **8** is a block diagram showing a logical view of an inter-FPGA crossbar network according to one embodiment of the present invention. As described earlier, the inter-FPGA crossbar network **750** interconnects the FPGAs on a circuit board such that signals can be routed between any of the FPGAs on the circuit board. In addition, the inter-FPGA crossbar network **750** also interconnects the FPGAs to the circuit board I/O connections so that signals can be routed between the circuit board I/O connections and the FPGAs. The interconnection of logical signals is illustrated in FIG. **8**. As discussed above, the actual physical transfer of these signals is performed using the regional time multiplexing of the present invention.

The routing of signals in the inter-FPGA crossbar network **750** spans both the FPGA level and the circuit board level. A division line **700** is shown in FIG. **8** which identifies a separation between FPGA level **701** and board level **702**. Crossbars **230** (corresponding to stage **114***a*–/**114***b*) is implemented in FPGA **610** of FIG. **4**. The second stage of the inter-FPGA crossbar network, however, is implemented in the board level **702**. I/O signals **(16)** from each of the four crossbars **230** of the 24 FPGAs (4×24=96) I/O coupled to the "FPGA-side" of the 16 RCs **631**. On the "board-side" of 16 RCs **631**, 28×16=448 signals are coupled to and from the logic board's I/O connections **640**.

Multiple signal routing time domains are also illustrated in FIG. **8**. The outputs of the FPGAs, from crossbars **230**, are in signal routing time domain(s) **703**. As discussed above, different sets of I/Os from an FPGA, or different FPGAs, can be in different signal routing time domains. Similarly, the RCs **631** are in signal routing time domain(s) **704**. As discussed above, different sets of I/Os from an RC, or different RCs, can be in different signal routing time domains.

For the above described embodiment, wherein there are 24 FPGAs **610**, each having 64 I/O connections, disposed on circuit board **600**, having 448 I/O connections, a total of {(24×64)+448} or {1536+448} are interconnected together by inter-FPGA crossbar network **750**.

FIG. **9** is a block diagram of a backplane assembly according to one embodiment of the present invention. Backplane assembly **800** is used to interconnect circuit boards **820**. Circuit boards **820** may be logic boards **600** of FIG. **4** or I/O boards for interfacing with external devices. In other words, backplane assembly **800** is used to interconnect FPGAs disposed on logic boards **600** with each other and with external devices. Backplane assembly **800** comprises backplane **810** and a number of matrix boards **835**. Backplane **810** is used to accept circuit boards **820**, whereas matrix boards **835** are used to interconnect signals to and from the various circuit boards **820**.

Recall from earlier descriptions that inter-FPGA crossbar network stage **1** also "doubles up" as inter-board crossbar network stage 0. Collectively, the 16 sets of 28 RCs **805** implement inter-board crossbar network stage 1. Together, the two stages implement the inter-board crossbar network. As discussed above, different sets of I/Os of the RCs **620** and **837** can be in different time domains. Additionally, each of the crossbar **1010** inputs/outputs can be configured with the regional time multiplexing circuitry of the present invention.

FIG. **10** is a block diagram illustrating a logical view of an inter-board crossbar network according to one embodiment of the present invention. As described earlier, inter-board crossbar network **905** spans two physical form levels, i.e. board level **702** and backplane level **902** separated by dotted line **900**. As shown, for the illustrated embodiment, stage 0 comprises 23 124×124 crossbars **631**, each having 28 board I/O connections, whereas stage 1 comprises 28 27×27 crossbars **940**, each having 23 "board-side" I/O connections. The **28** board I/O connections of the 23 crossbars **631** and the 23 "board-side" I/O connections of the 28 crossbars **940** are connected to each other in a "butterfly" manner. Additionally, each crossbar **940** also has 4 "crate-side" I/O connections. The interconnection of logical signals is illustrated in FIG. **10**. As discussed above, the actual physical transfer of these signals is performed using the regional time multiplexing of the present invention.

Signals are transferred between two chips of the emulator described above via physical connections between those chips. According to one embodiment of the present invention, each of the physical connections allows for concurrent bi-directional data transfer. FIG. **11** is a block diagram illustrating the concurrent bi-directional data transfer over a single connection according to one embodiment of the present invention. As illustrated, two chips **1102** and **1104** are connected via a connection **1108**. Connection **1108** is intended to represent a wide range of conventional connection media, including both wires and circuit board traces. According to one embodiment of the present invention, the FPGAs and RCs discussed above are connected together analogously to chips **1102** and **1104**. For ease of explanation, only a single connection between two chips is illustrated. It is to be appreciated that additional signals can also be transferred between the chips in an analogous manner.

Chips **1102** and **1104** can simultaneously transfer signals to each other via connection **1108**. Chips **1102** and **1104** each include I/O circuitry, including a driver and a detection logic as illustrated. An output signal **1121** to be output by chip **1102** is driven onto connection **1108** via driver **1123**. Concurrently, an output signal **1132** to be output by chip **1104** is driven onto connection **1108** via driver **1133**. After the signals are driven onto connection **1108**, detection logics **1125** and **1135** each sample the voltage level of connection **1108**. Based on the sampled voltage level of connection **1108**, as well as possibly the output signal **1132**, detection logic **1135** provides an input signal **1131** to the internal circuitry of chip **1104** which is representative of output signal **1121** driven by chip **1102**. Similarly, based on the sampled voltage level of connection **1108**, as well as possibly the output signal **1121**, detection logic **1125** provides an input signal **1122** to the internal circuitry of chip **1102** which is representative of output signal **1132** driven by chip **1104**.

As is well-known to those skilled in the art, driving a particular value onto a connection is done by asserting a particular voltage level on the connection. A value of a logical zero is typically in the range of 0.0 volts to 0.5 volts, and the value of a logical one is typically in the range of 1.8

volts to 2.4 volts. Detection logics **1125** and **1135** use these voltage ranges in part to determine the value being driven by the other chip according to the following procedure. If the voltage level of connection **1108** is less than 0.5 volts, then both chips were driving a logical zero. If the voltage level of connection **1108** is greater than 1.8 volts, then both chips were driving a logical one. However, if the voltage level is between 0.5 volts and 1.8 volts, then one of the chips was driving a logical zero while the other was driving a logical one. As illustrated, detection logics **1125** and **1135** both receive as inputs the output signals being driven by their respective chips. According to the present invention, detection logic **1125** can, in the situation of a voltage level on connection **1108** between 0.5 volts and 1.8 volts, conclude that the signal output by chip **1104** is the inverse of the signal being output by chip **1102**. Similarly, detection logic **1135** can, in the situation of a voltage level on connection **1108** between 0.5 volts and 1.8 volts, conclude that the signal output by chip **1102** is the inverse of the signal being output by chip **1104**.

Thus, a single physical connection between two chips can be used to simultaneously transfer signals bi-directionally between those chips.

In the discussions above the regional time multiplexing is described as using two-to-one multiplexing, with two logical connections corresponding to one physical connection. Alternate embodiments of the present invention can use different numbers of inputs and outputs for the multiplexing, with m physical connections corresponding to n logical connections, where n>m, using an n to m multiplexer.

In the discussion above, the emulator is described as including multiple FPGAs. In alternate embodiments, other reconfigurable logic devices are used in the emulator rather than FPGAs.

Also in the discussions above, reference is made to chips which include pins. It is to be appreciated that the present invention can also be practiced in embodiments where chips do not include pins, such as where chips are surface mounted to circuit boards.

Thus, by separating the emulator into different regions, each being a separate time domain, asynchronous logic may be emulated without hard-wiring asynchronous signals to dedicated pins. Additionally, the problem of synchronizing clock signals is advantageously reduced, regardless of the overall size of the emulator. By not requiring the same clock signal to be routed throughout the entire system, the clock signals in the emulator no longer need to be synchronized across such a large area.

While the emulation system of the present invention has been described in terms of the above illustrated embodiments, those skilled in the art will recognize that the invention is not limited to the embodiments described. The present invention can be practiced with modification and altercation within the spirit and scope of the appended claims. The description is thus regarded as illustrative instead of restrictive on the present invention.

What is claimed is:

1. An emulation system comprising:

a first plurality of reconfigurable logic devices;

a second plurality of reconfigurable logic devices;

a third plurality of reconfigurable logic devices;

a first time multiplexed interconnection coupled to and situated between the first plurality of reconfigurable logic devices and the second plurality of reconfigurable logic devices; and

a second time multiplexed interconnection coupled to and situated between the second plurality of reconfigurable logic devices and the third plurality of reconfigurable logic devices, wherein clocking of the second time multiplexed interconnection is independent of clocking of the first time multiplexed interconnection.

2. The emulation system of claim **1**, wherein each of the first plurality of reconfigurable logic devices, each of the second plurality of reconfigurable logic devices, and each of the third plurality of reconfigurable logic devices is a field programmable gate array.

3. The emulation system of claim **1**, wherein the first time multiplexed interconnection includes a first set of input/output circuitry of a multi-clocked routing chip and the second time multiplexed interconnection includes a second set of input/output circuitry of the multi-clocked routing chip.

4. The emulation system of claim **1**, wherein the first time multiplexed interconnection includes a first plurality of multiplexers and demultiplexers, and the second time multiplexed interconnection includes a second plurality of multiplexers and demultiplexers.

5. An emulator for emulating a circuit design, comprising:

a first reconfigurable logic device that includes a first plurality of reconfigurable logic elements and first input/output circuitry;

a second reconfigurable logic device that includes a second plurality of reconfigurable logic elements and second input/output circuitry;

a first clock signal for clocking the first plurality of reconfigurable logic elements;

a second clock signal for clocking the second plurality of reconfigurable logic elements; and

at least one signal routing clock signal for clocking at least one of the first input/output circuitry and the second input/output circuitry, wherein the signal routing clock signal is independent of the first clock signal and the second clock signal.

6. An emulator according to claim **5**, wherein the first reconfigurable logic device is a field programmable gate array.

7. An emulator according to claim **6**, wherein the second reconfigurable logic device is a field programmable gate array.

8. An emulator according to claim **5**, wherein the first clock signal and the second clock signal are the same clock signal.

9. An emulator according to claim **5**, wherein said at least one signal routing clock signal includes a first signal routing clock signal for clocking the first input/output circuitry and a second signal routing clock signal for clocking the second input/output circuitry, wherein the first signal routing clock signal is different from the second signal routing clock signal.

10. An emulator according to claim **5**, further comprising:

at least one interconnect device interconnecting the first reconfigurable logic device and the second reconfigurable logic device, wherein said interconnect device includes a first input/output portion and a second input/output portion, wherein the first input/output circuitry and the first input/output portion are clocked by a first signal routing clock signal, and wherein the second input/output circuitry and the second input/output portion are clocked by a second signal routing clock signal.

11. An emulator according to claim **10**, wherein said at least one interconnect device includes a plurality of multi-

plexers for time multiplexing data transfers to and from another interconnect device.

12. An emulator according to claim **10**, wherein signals are transferred out of said interconnect device asynchronously to transfer of signals into said interconnect device.

13. An emulator according to claim **5**, further comprising:

a third reconfigurable logic device clocked at least in part by a third clock signal that is different from the first clock signal and the second clock signal.

14. An emulator according to claim **5**, further comprising:

a first interconnect device interconnecting the first reconfigurable logic device and the second reconfigurable logic device; and

a bi-directional data transfer connection situated between the first reconfigurable logic device and the first interconnect device, wherein the bi-directional data transfer connection provides simultaneous bi-directional data transfer between the first reconfigurable logic device and the first interconnect device via a single wire or trace.

15. An emulator according to claim **14**, wherein the first reconfigurable logic device includes detection logic for determining a signal value asserted by the first interconnect device based at least in part on a voltage level of the bi-directional data transfer connection.

16. An emulator according to claim **15**, wherein the detection logic determines the signal value asserted by the first interconnect device further based at least in part on a signal value asserted by the first reconfigurable logic device.

17. An emulator according to claim **5**, further comprising:

a first set of input/output pins connected to the first input/output circuitry for transferring signals to and from the first reconfigurable logic device; and

a second set of input/output pins connected to the second input/output circuitry for transferring signals to and from the second reconfigurable logic device.

18. An emulator according to claim **5**, wherein signals can be transferred into and out of the first reconfigurable logic device asynchronously to changing of signals internal to the first reconfigurable logic device.

19. An emulator according to claim **18**, wherein signals can be transferred into and out of the second reconfigurable logic device asynchronously to changing of signals internal to the second reconfigurable logic device.

20. An emulator according to claim **5**, wherein the first reconfigurable logic device is a first field programmable gate array located on a first chip and the second reconfigurable logic device is a second field programmable gate array located on a second chip separate from the first chip.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

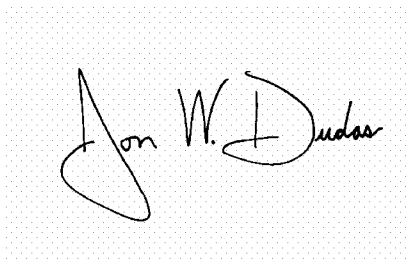
PATENT NO. : 6,947,882 B1
APPLICATION NO. : 09/404920
DATED : September 20, 2005
INVENTOR(S) : Frederic Reblewski et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Title page,
Item [75], Inventors, replace "Lepaps" with -- Lepape --.
Item [56], **References Cited**,
U.S. PATENT DOCUMENTS, insert the following:
-- 5,140,193 08/1992 Freeman
5,943,490 08/1999 Sample --.
FOREIGN PATENT DOCUMENTS, insert the following:
-- JP 04-138569 05/1992
JP 08-030653 02/1996
JP 08-508599 10/1996
JP 11-073440 03/1999 --.

Signed and Sealed this

Eleventh Day of July, 2006



JON W. DUDAS
Director of the United States Patent and Trademark Office

Minutes of Proceeding and Order,

Docket 686,

entered on Sept. 26, 2014

A137 - 143

From: info@ord.uscourts.gov
Sent: Friday, September 26, 2014 5:21 PM
To: nobody@ord.uscourts.gov
Subject: Activity in Case 3:10-cv-00954-MO Mentor Graphics Corporation v. EVE-USA, Inc. et al Order on Motion - Miscellaneous

This is an automatic e-mail message generated by the CM/ECF system. Please DO NOT RESPOND to this e-mail because the mail box is unattended.
*****NOTE TO PUBLIC ACCESS USERS*** Judicial Conference of the United States policy permits attorneys of record and parties in a case (including pro se litigants) to receive one free electronic copy of all documents filed electronically, if receipt is required by law or directed by the filer. PACER access fees apply to all other users. To avoid later charges, download a copy of each document during this first viewing. However, if the referenced document is a transcript, the free copy and 30 page limit do not apply.**

U.S. District Court

District of Oregon

Notice of Electronic Filing

The following transaction was entered on 9/26/2014 at 5:21 PM PDT and filed on 9/26/2014
Case Name: Mentor Graphics Corporation v. EVE-USA, Inc. et al
Case Number: 3:12-cv-01500-MO
Filer:
Document Number: 456(No document attached)

Docket Text:
MINUTES of Proceedings: Final Pretrial Conference held. Order GRANTING IN PART Motion in Limine #8 [627]. ORDER Donald Cantow was sufficiently disclosed as a damages witness with personal knowledge of Mentor Graphics's manufacturing capacity. Therefore Donald Cantow may testify at trial with respect to Mentor Graphics's manufacturing capacity. The parties are ordered to appear in the courtroom at 8:30am September 29, 2014 to determine the admissibility of Ms. Stuckwisch's damages testimony. The sole issue discussed will be whether or not Ms. Stuckwisch was properly disclosed as a potential damages witness for trial. Order GRANTING IN PART DENYING IN PART Motion in Limine [619]. Order DENYING #1; GRANTING #3. Motion to Exclude the Testimony of Stephen Degnan based on VirnetX, Inc. v. Cisco Systmes. Inc. [668] is DENIED. Mark Miller, Jeff Yost, George Riley, Anne E. Huffsmith and James E. Geringer present as counsel for plaintiff(s). Neal Chatterjee, Julia Markley, Steve English, Travis Jensen and Scott Lonardo present as counsel for defendant(s).(Court Reporter Dennis Apodaca.) Associated Cases: 3:10-cv-00954-MO, 3:12-cv-01500-MO, 3:13-cv-00579-MO(dls)

Case Name: Synopsys Inc et al v. Mentor Graphics Corporation
Case Number: 3:13-cv-00579-MO
Filer:

Document Number: 503(No document attached)

Docket Text:
MINUTES of Proceedings: Final Pretrial Conference held. Order GRANTING IN PART Motion in Limine #8 [627]. ORDER Donald Cantow was sufficiently disclosed as a damages witness with personal knowledge of Mentor Graphics's manufacturing capacity. Therefore Donald Cantow may testify at trial with respect to Mentor Graphics's manufacturing capacity. The parties are ordered to appear in the courtroom at 8:30am September 29, 2014 to determine the admissibility of Ms. Stuckwisch's damages testimony. The sole issue discussed will be whether or not Ms. Stuckwisch was properly disclosed as a potential damages witness for trial. Order GRANTING IN PART DENYING IN PART Motion in Limine [619]. Order DENYING #1; GRANTING #3. Motion to Exclude the Testimony of Stephen Degnan based on VirnetX, Inc. v. Cisco Systmes. Inc. [668] is DENIED. Mark Miller, Jeff Yost, George Riley, Anne E. Huffsmith and James E. Geringer present as counsel for plaintiff(s). Neal Chatterjee, Julia Markley, Steve English, Travis Jensen and Scott Lonardo present as counsel for defendant(s).(Court Reporter Dennis Apodaca.) Associated Cases: 3:10-cv-00954-MO, 3:12-cv-01500-MO, 3:13-cv-00579-MO(dls)

Case Name: Mentor Graphics Corporation v. EVE-USA, Inc. et al
Case Number: 3:10-cv-00954-MO
Filer:
Document Number: 686(No document attached)

Docket Text:
MINUTES of Proceedings: Final Pretrial Conference held. Order GRANTING IN PART Motion in Limine #8 [627]. ORDER Donald Cantow was sufficiently disclosed as a damages witness with personal knowledge of Mentor Graphics's manufacturing capacity. Therefore Donald Cantow may testify at trial with respect to Mentor Graphics's manufacturing capacity. The parties are ordered to appear in the courtroom at 8:30am September 29, 2014 to determine the admissibility of Ms. Stuckwisch's damages testimony. The sole issue discussed will be whether or not Ms. Stuckwisch was properly disclosed as a potential damages witness for trial. Order GRANTING IN PART DENYING IN PART Motion in Limine [619]. Order DENYING #1; GRANTING #3. Motion to Exclude the Testimony of Stephen Degnan based on VirnetX, Inc. v. Cisco Systmes. Inc. [668] is DENIED. Mark Miller, Jeff Yost, George Riley, Anne E. Huffsmith and James E. Geringer present as counsel for plaintiff(s). Neal Chatterjee, Julia Markley, Steve English, Travis Jensen and Scott Lonardo present as counsel for defendant(s).(Court Reporter Dennis Apodaca.) Associated Cases: 3:10-cv-00954-MO, 3:12-cv-01500-MO, 3:13-cv-00579-MO(dls)

3:12-cv-01500-MO Notice has been electronically mailed to:

Scott D. Eads seads@perkinscoie.com, MTompkins@perkinscoie.com, ceffenberger-legg@perkinscoie.com, docketpor@perkinscoie.com, khuang@perkinscoie.com

Stephen F. English senglish@perkinscoie.com, ajaclark@perkinscoie.com

Dennis P. Rawlinson (Terminated) dennis.rawlinson@millernash.com, jane.rausch@millernash.com

James E. Geringer james.geringer@klarquist.com, amy.uhl@klarquist.com, toni.kammers@klarquist.com

James C. Brooks jbrooks@orrick.com

Kristin L. Cleveland kristin.cleveland@klarquist.com, julia.castellano@klarquist.com, litigationdocketing@klarquist.com, mary.pressel@klarquist.com

Julia E. Markley jmarkley@perkinscoie.com, ceffenberger-legg@perkinscoie.com, docketpor@perkinscoie.com, jgilliland@perkinscoie.com

Patrick M. Bible patrick.bible@klarquist.com

Salumeh R. Loesch salumeh.loesch@klarquist.com, linda.schroeder@klarquist.com

George Riley griley@omm.com

Michael Sapoznikow msapoznikow@omm.com, cweiser@omm.com, wbenson@omm.com

Mark E. Miller markmiller@omm.com

Mark W. Wilson mark.wilson@klarquist.com

Andrew M. Mason andrew.mason@klarquist.com, linda.schroeder@klarquist.com, litigationdocketing@klarquist.com

Gino Cheng gcheng@orrick.com

Robin M. Wall rwall@omm.com

Travis Jensen tjensen@orrick.com

Scott Lonardo slonardo@orrick.com, aako-nai@orrick.com, cphan@orrick.com

David T. Pritikin dpritikin@sidley.com

M. Patricia Thayer pthayer@sidley.com, sfdocket@sidley.com

Philip W. Woo pwoo@sidley.com, mhanhan@sidley.com

Aseem Saran Gupta agupta@sidley.com, cchi@sidley.com, crowland@sidley.com, rlawson@sidley.com, rpask@sidley.com, tchandler@sidley.com

David T. DeZern ddezern@sidley.com

Nick Janda njanda@omm.com

Geoffrey H. Yost gyost@omm.com

Nicole J. Lindquist nicole.lindquist@wilmerhale.com, crystal.roberts@wilmerhale.com, gregory.lantier@wilmerhale.com, liv.herriot@wilmerhale.com

Elysa Q. Wan ewan@omm.com

Marc J. Pensabene mpensabene@omm.com

Richard F. Martinelli rmartinelli@orrick.com

Anne E. Huffsmith ahuffsmith@omm.com

Michael Koplow mkoplow@omm.com

3:12-cv-01500-MO Notice will not be electronically mailed to:

3:13-cv-00579-MO Notice has been electronically mailed to:

Scott D. Eads seads@perkinscoie.com, MTompkins@perkinscoie.com, ceffenberger-legg@perkinscoie.com, docketpor@perkinscoie.com, khuang@perkinscoie.com

Stephen F. English senglish@perkinscoie.com, ajaclark@perkinscoie.com

Dennis P. Rawlinson (Terminated) dennis.rawlinson@millernash.com, jane.rausch@millernash.com

James E. Geringer james.geringer@klarquist.com, amy.uhl@klarquist.com, toni.kammers@klarquist.com

Julia E. Markley jmarkley@perkinscoie.com, ceffenberger-legg@perkinscoie.com, docketpor@perkinscoie.com, jgilliland@perkinscoie.com, twoolley@perkinscoie.com

Vickie L. Feeman vfeeman@orrick.com

Patrick M. Bible patrick.bible@klarquist.com

Salumeh R. Loesch salumeh.loesch@klarquist.com, linda.schroeder@klarquist.com

George Riley griley@omm.com

Michael Sapoznikow msapoznikow@omm.com, cweiser@omm.com, wbenson@omm.com

Mark E. Miller markmiller@omm.com

Ricardo Rodriguez (Terminated) douglasjar@cooley.com, elliottds@cooley.com

Mark W. Wilson mark.wilson@klarquist.com

Douglas E. Lumish (Terminated) doug.lumish@lw.com, #svlitigationservices@lw.com, arlene.uyemura@lw.com

Gabriel S. Gross (Terminated) gabe.gross@lw.com, martha.rubio@lw.com

Jeffrey G. Homrig (Terminated) jeff.homrig@lw.com, #svlitigationservices@lw.com, martha.rubio@lw.com

Gino Cheng gcheng@orrick.com

Indra Neel Chatterjee nchatterjee@orrick.com, kmudurian@orrick.com

William H. Wright wwright@orrick.com, alchou@orrick.com, pjohnsen@orrick.com

Jesse Y. Cheng jcheng@orrick.com, jdeforest@orrick.com, kbarnick@orrick.com, sstillman@orrick.com

Scott D. Lindlaw slindlaw@orrick.com, descamilla@orrick.com

Robin M. Wall rwall@omm.com

Travis Jensen tjensen@orrick.com

Scott Lonardo slonardo@orrick.com, aako-nai@orrick.com, cphan@orrick.com

David T. Pritikin dpritikin@sidley.com

M. Patricia Thayer pthayer@sidley.com, sfdocket@sidley.com

Philip W. Woo pwoo@sidley.com, mhanhan@sidley.com

Aseem Saran Gupta agupta@sidley.com, cchi@sidley.com, crowland@sidley.com, rlawson@sidley.com, rpask@sidley.com, tchandler@sidley.com

David T. DeZern ddezern@sidley.com

Nick Janda njanda@omm.com

Geoffrey H. Yost gyost@omm.com

Nicole J. Lindquist nicole.lindquist@wilmerhale.com, crystal.roberts@wilmerhale.com, gregory.lantier@wilmerhale.com, liv.herriot@wilmerhale.com

Elysa Q. Wan ewan@omm.com

Marc J. Pensabene mpensabene@omm.com

Richard F. Martinelli rmartinelli@orrick.com

Anne E. Huffsmith ahuffsmith@omm.com

Michael Koplow mkoplow@omm.com

3:13-cv-00579-MO Notice will <u>not</u> be electronically mailed to:

Christopher Ryan Ottenweller
Orrick, Herrington & Sutcliffe
1000 Marsh road
Menlo Park, CA 94025

Diana Catherine Rogosa
O'Melveny and Myers LLP
Two Embarcadero Center

28th Floor
San Francisco, CA 94565

James C. Brooks
Orrick Herrington & Sutcliffe, LLP
777 South Figueroa Street
Suite 3200
Los Angeles, CA 90017-5855

Robert John Benson
ORRICK, HERRINGTON & SUTCLIFFE LLP
1000 Marsh Road
Menlo Park, CA 94025

3:10-cv-00954-MO Notice has been electronically mailed to:

Scott D. Eads seads@perkinscoie.com, MTompkins@perkinscoie.com, ceffenberger-legg@perkinscoie.com, docketpor@perkinscoie.com, khuang@perkinscoie.com

Stephen F. English senglish@perkinscoie.com, ajaclark@perkinscoie.com

Dennis P. Rawlinson (Terminated) dennis.rawlinson@millernash.com, jane.rausch@millernash.com

James E. Geringer james.geringer@klarquist.com, amy.uhl@klarquist.com, toni.kammers@klarquist.com

James C. Brooks jbrooks@orrick.com

Kristin L. Cleveland kristin.cleveland@klarquist.com, julia.castellano@klarquist.com, litigationdocketing@klarquist.com, mary.pressel@klarquist.com

Julia E. Markley jmarkley@perkinscoie.com, ceffenberger-legg@perkinscoie.com, docketpor@perkinscoie.com, jgilliland@perkinscoie.com, twoolley@perkinscoie.com

Vickie L. Feeman vfeeman@orrick.com

Patrick M. Bible patrick.bible@klarquist.com

Salumeh R. Loesch salumeh.loesch@klarquist.com, linda.schroeder@klarquist.com

George Riley griley@omm.com

Luann Simmons lsimmons@omm.com

Michael Sapoznikow msapoznikow@omm.com, cweiser@omm.com, wbenson@omm.com

Mark E. Miller markmiller@omm.com

Mark W. Wilson mark.wilson@klarquist.com

Andrew M. Mason andrew.mason@klarquist.com, linda.schroeder@klarquist.com, litigationdocketing@klarquist.com

Indra Neel Chatterjee nchatterjee@orrick.com, kmudurian@orrick.com

William H. Wright wwright@orrick.com, Shanee.Nelson@synopsys.com, alchou@orrick.com, pjohnsen@orrick.com

Jesse Y. Cheng jcheng@orrick.com, jdeforest@orrick.com, kbarnick@orrick.com, sstillman@orrick.com

Scott D. Lindlaw slindlaw@orrick.com, descamilla@orrick.com

Robin M. Wall rwall@omm.com

Travis Jensen tjensen@orrick.com

Scott Lonardo slonardo@orrick.com, aako-nai@orrick.com, cphan@orrick.com

David T. Pritikin dpritikin@sidley.com

M. Patricia Thayer pthayer@sidley.com, sfdocket@sidley.com

Philip W. Woo pwoo@sidley.com, mhanhan@sidley.com

Aseem Saran Gupta agupta@sidley.com, cchi@sidley.com, crowland@sidley.com, rlawson@sidley.com, rpask@sidley.com, tchandler@sidley.com

David T. DeZern ddezern@sidley.com

Nick Janda njanda@omm.com

Geoffrey H. Yost gyost@omm.com

Nicole J. Lindquist nicole.lindquist@wilmerhale.com, crystal.roberts@wilmerhale.com, gregory.lantier@wilmerhale.com, liv.herriot@wilmerhale.com

Elysa Q. Wan ewan@omm.com

Leonard Joseph Denbina (Terminated) ldenbina@sidley.com

Marc J. Pensabene mpensabene@omm.com

Richard F. Martinelli rmartinelli@orrick.com

Anne E. Huffsmith ahuffsmith@omm.com

Michael Koplow mkoplow@omm.com

3:10-cv-00954-MO Notice will <u>not</u> be electronically mailed to:

Opinion and Order, Docket 524, signed by Judge Mosman on June 4, 2014 A14060 - 14062

UNITED STATES DISTRICT COURT

DISTRICT OF OREGON

PORTLAND DIVISION

MENTOR GRAPHICS CORPORATION, an Oregon Corporation,

Plaintiff/Counter-defendant,

v.

EVE-USA, INC., a Delaware corporation; and **SYNOPSYS EMULATION AND VERIFICATION S.A.**, formed under the laws of France,

Defendants/Counter-claimants.

EVE-USA, INC., a Delaware corporation; and **SYNOPSYS EMULATION AND VERIFICATION S.A.**, formed under the laws of France,

Plaintiffs/Counter-defendants

v.

MENTOR GRAPHICS CORPORATION, an Oregon corporation,

Defendant/Counter-claimant.

Case No. 3:10-cv-954-MO (lead)
Case No. 3:12-cv-1500-MO
Case No. 3:13-cv-579-MO

OPINION AND ORDER

MOSMAN, J.,

EVE-USA, Inc., and Synopsys Emulation and Verification S.A. (collectively, "Synopsys") move for partial summary judgment on Mentor Graphics Corporation's ("Mentor Graphics") counterclaims in case no. 13-579 on grounds of claim preclusion. (Mot. [371] at 2.)[1] I took the motion under advisement [416] after hearing oral argument on September 24, 2013. On March 5, 2014, I ordered [482] supplemental briefing on the question whether the ZeBu emulators at issue in Mentor Graphics's counterclaims are "essentially the same" as those it accused of patent infringement in a 2006 action against EVE-USA, Inc. Mentor Graphics's supplemental brief also raises the issue of whether, in light of recent Federal Circuit opinions, the claim preclusion inquiry no longer turns on whether the presently and formerly accused products are "essentially the same." (Supp. Opp. [490] at 2–4.) I now GRANT Synopsys's motion.

For at least two decades, the Federal Circuit has held that judgment on a claim of infringement against a product bars a later infringement claim against a different product if the two products are "essentially the same." *Nystrom v. Trex Co., Inc.*, 580 F.3d 1281, 1285 (Fed. Cir. 2009); *Roche Palo Alto LLC v. Apotex, Inc.*, 531 F.3d 1372, 1379 (Fed. Cir. 2008); *Foster v. Hallco Mfg. Co., Inc.*, 947 F.2d 469, 479–80 (Fed. Cir. 1991). By contrast, in two recent, three-judge panel opinions, the court has held that claim preclusion does not bar an infringement claim against a product that did not exist as of the prior judgment, "essentially the same" or not. *Brain Life, LLC v. Elekta Inc.*, 746 F.3d 1045, 1054 (Fed. Cir. 2014); *Aspex Eyewear, Inc. v. Marchon Eyewear, Inc.*, 672 F.3d 1335, 1342 (Fed. Cir. 2012). Synopsys correctly observes that the *Foster* line of cases cannot be reconciled with *Aspex* and *Brain Life*. (Supp. Resp. [502] at 2–3.) Because one three-judge panel cannot overrule another, *Newell Cos., Inc. v. Kenny Mfg. Co.*, 864 F.2d 757, 765 (Fed. Cir. 1988), *Foster* controls until the Federal Circuit sitting *en banc* says

[1] All docket numbers refer to the lead case, no. 10-954.

otherwise. Mentor Graphics's counterclaims are barred if they accuse products that are "essentially the same" as those accused in the 2006 action.

The evidence that the parties have presented permits only one conclusion: the presently and formerly accused ZeBu emulators are "essentially the same." Mentor Graphics identifies only two new features of the modern emulators: a user-friendly software tool for generating "transactors" called ZEMI-3 and a device allowing multiple ZeBu units to function as a single emulator called Fast Internal Bus. (Supp. Opp. [490] at 6–9.) Synopsys correctly observes that neither of these features materially alters the manner in which the newer ZeBu emulators allegedly practice Mentor Graphics's patents. (Supp. Resp. [502] at 6–9.)

Synopsys's Motion for Partial Summary Judgment [371] is GRANTED. Claim preclusion bars Mentor Graphics's counterclaims [382] to the extent that they allege infringement of U.S. Patent Nos. 6,009,531 and 5,649,176.

IT IS SO ORDERED.

DATED this 4th day of June, 2014.

/s/ Michael W. Mosman
MICHAEL W. MOSMAN
United States District Judge

Minutes of Proceeding and Order,

Docket 582,

entered on July 29, 2014

A23748 - 23756

From: info@ord.uscourts.gov
Sent: Tuesday, August 05, 2014 2:53 PM
To: nobody@ord.uscourts.gov
Subject: Activity in Case 3:10-cv-00954-MO Mentor Graphics Corporation v. EVE-USA, Inc. et al Order on motion for summary judgment

This is an automatic e-mail message generated by the CM/ECF system. Please DO NOT RESPOND to this e-mail because the mail box is unattended.
*****NOTE TO PUBLIC ACCESS USERS*** Judicial Conference of the United States policy permits attorneys of record and parties in a case (including pro se litigants) to receive one free electronic copy of all documents filed electronically, if receipt is required by law or directed by the filer. PACER access fees apply to all other users. To avoid later charges, download a copy of each document during this first viewing. However, if the referenced document is a transcript, the free copy and 30 page limit do not apply.**

U.S. District Court

District of Oregon

Notice of Electronic Filing

The following transaction was entered on 07/29/2014 at 2:46:33 PM PDT and filed on 07/28/2014
Case Name: Mentor Graphics Corporation v. EVE-USA, Inc. et al
Case Number: 3:10-cv-00954-MO
Filer:
Document Number: 582 (No document attached)

Docket Text:
MINUTES of Proceedings: Motion Hearing Held. Synopsyss Motion To Strike Portions of Mentors Experts Reply Reports [533] is GRANTED IN PART and DENIED IN PART. The motion is DENIED as to Dr. Sarrafzadehs Reply Expert Report. The motion is GRANTED as to the Reply Expert Reports of Dr. Degnan and Ms. Stuckwisch. Accordingly, paragraphs 3132 of Dr. Degnans Reply Expert Report and paragraph 93 of Ms. Stuckwischs Reply Expert Report are hereby stricken. Mentor Graphicss Motion for Summary Judgment [531] is GRANTED IN PART and DENIED IN PART as follows. The motion is GRANTED as to whether CoBALT anticipates or renders obvious claims 3 and 6 of U.S. Patent No. 6,876,962. Synopsys has not presented sufficient evidence to permit an inference that the ET3 chip is an FPGA, that replacing CoBALTs emulation processors with FPGAs would be obvious, or that the partitioning step of the ET3 compilation flow generates netlists. The motion is DENIED as to whether the Jean and Burns references render the 962 Patent obvious. A jury might find that a person of skill in the art reading both references together would know that they could be adapted to construct the emulation system described in the 962 Patent. The motion is DENIED AS MOOT with respect to infringement of claims 7 and 9 of the 882 Patent based on the Courts disposition of Synopsyss motion. Synopsyss Motion for Summary Judgment [537] is GRANTED IN PART, DENIED IN PART, and TAKEN UNDER ADVISEMENT as follows. The motion is GRANTED with respect to invalidity of claims 7, 9, and 13 of U.S. Patent No. 6,947,882. The 882 Patents

specification describes the minimum frequency relationship between the signal routing clock signal and the first and second clock signals as an exception to independent clocking. As a result, the specification does not demonstrate possession of the unqualifiedly independent clocking that the asserted claims require, and the claims do not meet the written description requirement. Synopsyss motion is DENIED AS MOOT with respect to availability of prefiling damages for infringement of the 882 Patent. The motion is GRANTED as to noninfringement of claims 3, 5, and 6 of U.S. Patent No. 6,876,962" is amended to read "The motion is GRANTED as to noninfringement of claims 3, 5, 6, and 8 of U.S. Patent No. 6,876,962. Mentor Graphicss theory of infringement fails as a matter of law because the claims require the computer itself to take into account the resources allocated or not allocated to a first design when generating a netlist for a second design. Moreover, even if Mentor Graphicss infringement theory were cognizable, it has presented no evidence that Synopsys or EVE ever instructed a customer to configure a second design based on the resources tied up in a first design, or that any customer ever did so. Such direct evidence of specific instances of infringement is required under ACCO Brands, Inc. v. ABA Locks Mfr. Co., Ltd., 501 F.3d 1307 (Fed. Cir. 2012), where, as here, the alleged infringer has not instructed its customers in the allegedly infringing use of its product. The direct evidence Mentor Graphics produced concerned reconfiguring a first design in order to maximize the space available for other designs, not configuring a second design taking into consideration the emulation resources allocated or not allocated to the first design. Synopsyss motion is taken under advisement as to recoverability of lost profits. George Riley, Mark E. Miller and Michael Sapoznikow present as counsel for plaintiff(s). Stephen English, Indra Neel Chatterjee and Scott Lonardo present as counsel for defendant(s).(Court Reporter Bonita Shumway.) Associated Cases: 3:10-cv-00954-MO, 3:12-cv-01500-MO, 3:13-cv-00579-MO(dls) Modified on 8/5/2014 (dls).

Case Name: Synopsys Inc et al v. Mentor Graphics Corporation
Case Number: 3:13-cv-00579-MO
Filer:
Document Number: 401 (No document attached)

Docket Text:
MINUTES of Proceedings: Motion Hearing Held. Synopsyss Motion To Strike Portions of Mentors Experts Reply Reports [533] is GRANTED IN PART and DENIED IN PART. The motion is DENIED as to Dr. Sarrafzadehs Reply Expert Report. The motion is GRANTED as to the Reply Expert Reports of Dr. Degnan and Ms. Stuckwisch. Accordingly, paragraphs 3132 of Dr. Degnans Reply Expert Report and paragraph 93 of Ms. Stuckwischs Reply Expert Report are hereby stricken. Mentor Graphicss Motion for Summary Judgment [531] is GRANTED IN PART and DENIED IN PART as follows. The motion is GRANTED as to whether CoBALT anticipates or renders obvious claims 3 and 6 of U.S. Patent No. 6,876,962. Synopsys has not presented sufficient evidence to permit an inference that the ET3 chip is an FPGA, that replacing CoBALTs emulation processors with FPGAs would be obvious, or that the partitioning step of the ET3 compilation flow generates netlists. The motion is DENIED as to whether the Jean and Burns references render the 962 Patent obvious. A jury might find that a person of skill in the art reading both references together would know that they could be adapted to construct the emulation system described in the 962 Patent. The motion is DENIED AS MOOT with respect to infringement of claims 7 and 9 of the 882 Patent based on the Courts disposition of Synopsyss motion. Synopsyss Motion for Summary Judgment [537] is GRANTED IN PART, DENIED IN PART, and TAKEN UNDER ADVISEMENT as follows. The motion is GRANTED with respect to invalidity of claims 7, 9, and 13 of U.S. Patent No. 6,947,882. The 882 Patents specification describes the minimum frequency relationship between the signal routing clock

signal and the first and second clock signals as an exception to independent clocking. As a result, the specification does not demonstrate possession of the unqualifiedly independent clocking that the asserted claims require, and the claims do not meet the written description requirement. Synopsyss motion is DENIED AS MOOT with respect to availability of prefiling damages for infringement of the 882 Patent. The motion is GRANTED as to noninfringement of claims 3, 5, and 6 of U.S. Patent No. 6,876,962" is amended to read "The motion is GRANTED as to noninfringement of claims 3, 5, 6, and 8 of U.S. Patent No. 6,876,962. Mentor Graphicss theory of infringement fails as a matter of law because the claims require the computer itself to take into account the resources allocated or not allocated to a first design when generating a netlist for a second design. Moreover, even if Mentor Graphicss infringement theory were cognizable, it has presented no evidence that Synopsys or EVE ever instructed a customer to configure a second design based on the resources tied up in a first design, or that any customer ever did so. Such direct evidence of specific instances of infringement is required under ACCO Brands, Inc. v. ABA Locks Mfr. Co., Ltd., 501 F.3d 1307 (Fed. Cir. 2012), where, as here, the alleged infringer has not instructed its customers in the allegedly infringing use of its product. The direct evidence Mentor Graphics produced concerned reconfiguring a first design in order to maximize the space available for other designs, not configuring a second design taking into consideration the emulation resources allocated or not allocated to the first design. Synopsyss motion is taken under advisement as to recoverability of lost profits. George Riley, Mark E. Miller and Michael Sapoznikow present as counsel for plaintiff(s). Stephen English, Indra Neel Chatterjee and Scott Lonardo present as counsel for defendant(s).(Court Reporter Bonita Shumway.) Associated Cases: 3:10-cv-00954-MO, 3:12-cv-01500-MO, 3:13-cv-00579-MO(dls) Modified on 8/5/2014 (dls).

Case Name: Mentor Graphics Corporation v. EVE-USA, Inc. et al
Case Number: 3:12-cv-01500-MO
Filer:
Document Number: 354 (No document attached)

Docket Text:
MINUTES of Proceedings: Motion Hearing Held. Synopsyss Motion To Strike Portions of Mentors Experts Reply Reports [533] is GRANTED IN PART and DENIED IN PART. The motion is DENIED as to Dr. Sarrafzadehs Reply Expert Report. The motion is GRANTED as to the Reply Expert Reports of Dr. Degnan and Ms. Stuckwisch. Accordingly, paragraphs 3132 of Dr. Degnans Reply Expert Report and paragraph 93 of Ms. Stuckwischs Reply Expert Report are hereby stricken. Mentor Graphicss Motion for Summary Judgment [531] is GRANTED IN PART and DENIED IN PART as follows. The motion is GRANTED as to whether CoBALT anticipates or renders obvious claims 3 and 6 of U.S. Patent No. 6,876,962. Synopsys has not presented sufficient evidence to permit an inference that the ET3 chip is an FPGA, that replacing CoBALTs emulation processors with FPGAs would be obvious, or that the partitioning step of the ET3 compilation flow generates netlists. The motion is DENIED as to whether the Jean and Burns references render the 962 Patent obvious. A jury might find that a person of skill in the art reading both references together would know that they could be adapted to construct the emulation system described in the 962 Patent. The motion is DENIED AS MOOT with respect to infringement of claims 7 and 9 of the 882 Patent based on the Courts disposition of Synopsyss motion. Synopsyss Motion for Summary Judgment [537] is GRANTED IN PART, DENIED IN PART, and TAKEN UNDER ADVISEMENT as follows. The motion is GRANTED with respect to invalidity of claims 7, 9, and 13 of U.S. Patent No. 6,947,882. The 882 Patents specification describes the minimum frequency relationship between the signal routing clock signal and the first and second clock signals as an exception to independent clocking. As a

result, the specification does not demonstrate possession of the unqualifiedly independent clocking that the asserted claims require, and the claims do not meet the written description requirement. Synopsyss motion is DENIED AS MOOT with respect to availability of prefiling damages for infringement of the 882 Patent. The motion is GRANTED as to noninfringement of claims 3, 5, and 6 of U.S. Patent No. 6,876,962" is amended to read "The motion is GRANTED as to noninfringement of claims 3, 5, 6, and 8 of U.S. Patent No. 6,876,962. Mentor Graphicss theory of infringement fails as a matter of law because the claims require the computer itself to take into account the resources allocated or not allocated to a first design when generating a netlist for a second design. Moreover, even if Mentor Graphicss infringement theory were cognizable, it has presented no evidence that Synopsys or EVE ever instructed a customer to configure a second design based on the resources tied up in a first design, or that any customer ever did so. Such direct evidence of specific instances of infringement is required under ACCO Brands, Inc. v. ABA Locks Mfr. Co., Ltd., 501 F.3d 1307 (Fed. Cir. 2012), where, as here, the alleged infringer has not instructed its customers in the allegedly infringing use of its product. The direct evidence Mentor Graphics produced concerned reconfiguring a first design in order to maximize the space available for other designs, not configuring a second design taking into consideration the emulation resources allocated or not allocated to the first design. Synopsyss motion is taken under advisement as to recoverability of lost profits. George Riley, Mark E. Miller and Michael Sapoznikow present as counsel for plaintiff(s). Stephen English, Indra Neel Chatterjee and Scott Lonardo present as counsel for defendant(s).(Court Reporter Bonita Shumway.) Associated Cases: 3:10-cv-00954-MO, 3:12-cv-01500-MO, 3:13-cv-00579-MO(dls) Modified on 8/5/2014 (dls).

3:10-cv-00954-MO Notice has been electronically mailed to:

Scott D. Eads seads@perkinscoie.com, ceffenberger-legg@perkinscoie.com, docketpor@perkinscoie.com, khuang@perkinscoie.com

Stephen F. English senglish@perkinscoie.com, ajaclark@perkinscoie.com

Dennis P. Rawlinson (Terminated) dennis.rawlinson@millernash.com, jane.rausch@millernash.com

James E. Geringer james.geringer@klarquist.com, toni.kammers@klarquist.com

James C. Brooks jbrooks@orrick.com

Kristin L. Cleveland kristin.cleveland@klarquist.com, julia.castellano@klarquist.com, litigationdocketing@klarquist.com, mary.pressel@klarquist.com

Julia E. Markley jmarkley@perkinscoie.com, asandvig@perkinscoie.com, ceffenberger-legg@perkinscoie.com, docketpor@perkinscoie.com, twoolley@perkinscoie.com

Vickie L. Feeman vfeeman@orrick.com, hrupert@orrick.com, nnurrenbern@orrick.com

Patrick M. Bible patrick.bible@klarquist.com

Salumeh R. Loesch salumeh.loesch@klarquist.com, linda.schroeder@klarquist.com

George Riley griley@omm.com

Luann Simmons lsimmons@omm.com

Michael Sapoznikow msapoznikow@omm.com, cweiser@omm.com, mkoplow@omm.com, wbenson@omm.com

Mark E. Miller markmiller@omm.com

Mark W. Wilson mark.wilson@klarquist.com

Andrew M. Mason andrew.mason@klarquist.com, linda.schroeder@klarquist.com, litigationdocketing@klarquist.com

Indra Neel Chatterjee nchatterjee@orrick.com, kmudurian@orrick.com

William H. Wright wwright@orrick.com, Shanee.Nelson@synopsys.com, alchou@orrick.com, hrupert@orrick.com, pjohnsen@orrick.com

Jesse Y. Cheng jcheng@orrick.com, jdeforest@orrick.com, kbarnick@orrick.com, sstillman@orrick.com

Scott D. Lindlaw slindlaw@orrick.com, descamilla@orrick.com

Robin M. Wall rwall@omm.com

Travis Jensen tjensen@orrick.com

Scott Lonardo slonardo@orrick.com, aako-nai@orrick.com, cphan@orrick.com

David T. Pritikin dpritikin@sidley.com

M. Patricia Thayer pthayer@sidley.com, sfdocket@sidley.com

Philip W. Woo pwoo@sidley.com

Aseem Saran Gupta agupta@sidley.com, cchi@sidley.com, crowland@sidley.com, ldenbina@sidley.com, rlawson@sidley.com, rpask@sidley.com, tchandler@sidley.com

David T. DeZern ddezern@sidley.com

Nick Janda njanda@omm.com

Geoffrey H. Yost gyost@omm.com

Elysa Q. Wan ewan@omm.com

Leonard Joseph Denbina (Terminated) ldenbina@sidley.com

Marc J. Pensabene mpensabene@omm.com

Richard F. Martinelli rmartinelli@orrick.com

3:10-cv-00954-MO Notice will <u>not</u> be electronically mailed to:

3:13-cv-00579-MO Notice has been electronically mailed to:

Scott D. Eads seads@perkinscoie.com, ceffenberger-legg@perkinscoie.com, docketpor@perkinscoie.com, khuang@perkinscoie.com

Stephen F. English senglish@perkinscoie.com, ajaclark@perkinscoie.com

Dennis P. Rawlinson (Terminated) dennis.rawlinson@millernash.com, jane.rausch@millernash.com

James E. Geringer james.geringer@klarquist.com, toni.kammers@klarquist.com

Julia E. Markley jmarkley@perkinscoie.com, asandvig@perkinscoie.com, ceffenberger-legg@perkinscoie.com, docketpor@perkinscoie.com, twoolley@perkinscoie.com

Vickie L. Feeman vfeeman@orrick.com, hrupert@orrick.com, nnurrenbern@orrick.com

Patrick M. Bible patrick.bible@klarquist.com

Salumeh R. Loesch salumeh.loesch@klarquist.com, linda.schroeder@klarquist.com

George Riley griley@omm.com

Michael Sapoznikow msapoznikow@omm.com, cweiser@omm.com, wbenson@omm.com

Mark E. Miller markmiller@omm.com

Ricardo Rodriguez (Terminated) douglasjar@cooley.com, elliottds@cooley.com

Mark W. Wilson mark.wilson@klarquist.com

Douglas E. Lumish (Terminated) doug.lumish@lw.com, #svlitigationservices@lw.com, arlene.uyemura@lw.com

Gabriel S. Gross (Terminated) gabe.gross@lw.com, martha.rubio@lw.com

Jeffrey G. Homrig (Terminated) jeff.homrig@lw.com, #svlitigationservices@lw.com, martha.rubio@lw.com

Gino Cheng gcheng@orrick.com

Indra Neel Chatterjee nchatterjee@orrick.com, kmudurian@orrick.com

William H. Wright wwright@orrick.com, alchou@orrick.com, hrupert@orrick.com, pjohnsen@orrick.com

Jesse Y. Cheng jcheng@orrick.com, jdeforest@orrick.com, kbarnick@orrick.com, sstillman@orrick.com

Scott D. Lindlaw slindlaw@orrick.com, descamilla@orrick.com

Robin M. Wall rwall@omm.com

Travis Jensen tjensen@orrick.com

Scott Lonardo slonardo@orrick.com, aako-nai@orrick.com, cphan@orrick.com

David T. Pritikin dpritikin@sidley.com

M. Patricia Thayer pthayer@sidley.com, sfdocket@sidley.com

Philip W. Woo pwoo@sidley.com

Aseem Saran Gupta agupta@sidley.com, cchi@sidley.com, crowland@sidley.com, ldenbina@sidley.com, rlawson@sidley.com, rpask@sidley.com, tchandler@sidley.com

David T. DeZern ddezern@sidley.com

Nick Janda njanda@omm.com

Geoffrey H. Yost gyost@omm.com

Elysa Q. Wan ewan@omm.com

Marc J. Pensabene mpensabene@omm.com

Richard F. Martinelli rmartinelli@orrick.com

3:13-cv-00579-MO Notice will <u>not</u> be electronically mailed to:

Christopher Ryan Ottenweller
Orrick, Herrington & Sutcliffe
1000 Marsh road
Menlo Park, CA 94025

Diana Catherine Rogosa
O'Melveny and Myers LLP
Two Embarcadero Center
28th Floor
San Francisco, CA 94565

James C. Brooks
Orrick Herrington & Sutcliffe, LLP
777 South Figueroa Street
Suite 3200
Los Angeles, CA 90017-5855

Robert John Benson
ORRICK, HERRINGTON & SUTCLIFFE LLP
1000 Marsh Road
Menlo Park, CA 94025

3:12-cv-01500-MO Notice has been electronically mailed to:

Scott D. Eads seads@perkinscoie.com, ceffenberger-legg@perkinscoie.com, docketpor@perkinscoie.com,

khuang@perkinscoie.com

Stephen F. English senglish@perkinscoie.com, ajaclark@perkinscoie.com

Dennis P. Rawlinson (Terminated) dennis.rawlinson@millernash.com, jane.rausch@millernash.com

James E. Geringer james.geringer@klarquist.com, toni.kammers@klarquist.com

James C. Brooks jbrooks@orrick.com

Kristin L. Cleveland kristin.cleveland@klarquist.com, julia.castellano@klarquist.com, litigationdocketing@klarquist.com, mary.pressel@klarquist.com

Julia E. Markley jmarkley@perkinscoie.com, asandvig@perkinscoie.com, ceffenberger-legg@perkinscoie.com, docketpor@perkinscoie.com

Patrick M. Bible patrick.bible@klarquist.com

Salumeh R. Loesch salumeh.loesch@klarquist.com, linda.schroeder@klarquist.com

George Riley griley@omm.com

Michael Sapoznikow msapoznikow@omm.com, cweiser@omm.com, wbenson@omm.com

Mark E. Miller markmiller@omm.com

Mark W. Wilson mark.wilson@klarquist.com

Andrew M. Mason andrew.mason@klarquist.com, linda.schroeder@klarquist.com, litigationdocketing@klarquist.com

Gino Cheng gcheng@orrick.com

Robin M. Wall rwall@omm.com

Travis Jensen tjensen@orrick.com

Scott Lonardo slonardo@orrick.com, aako-nai@orrick.com, cphan@orrick.com

David T. Pritikin dpritikin@sidley.com

M. Patricia Thayer pthayer@sidley.com, sfdocket@sidley.com

Philip W. Woo pwoo@sidley.com

Aseem Saran Gupta agupta@sidley.com, cchi@sidley.com, crowland@sidley.com, ldenbina@sidley.com, rlawson@sidley.com, rpask@sidley.com, tchandler@sidley.com

David T. DeZern ddezern@sidley.com

Nick Janda njanda@omm.com

Geoffrey H. Yost gyost@omm.com

Elysa Q. Wan ewan@omm.com

Marc J. Pensabene mpensabene@omm.com

Richard F. Martinelli rmartinelli@orrick.com

3:12-cv-01500-MO Notice will not be electronically mailed to:

The following document(s) are associated with this transaction:

This is a re-generated NEF. Created on 8/5/2014 at 2:53 PM PDT

CERTIFICATE OF SERVICE AND FILING

I certify that on September 30, 2015, I electronically filed the foregoing, BRIEF AND ADDENDUM OF APPELLEE AND CROSS-APPELLANT MENTOR GRAPHICS CORPORATION, by using the Court's CM/ECF filing system.

I certify that all participants in the case are registered CM/ECF users and that service will be accomplished by the appellate CM/ECF system.

Dated: September 30, 2015

Respectfully submitted,

/s/ *Mark E. Miller*

Mark E. Miller
O'Melveny & Myers LLP
Two Embarcadero Center, 28th Floor
San Francisco, California 94111
Telephone: (415) 984-8700
Facsimile: (415) 984-8701
markmiller@omm.com

Counsel for Plaintiff-Cross-Appellant
Mentor Graphics Corp

CERTIFICATE OF COMPLIANCE

Counsel for Plaintiff-Cross-Appellant Mentor Graphics Corporation certifies:

1. This brief complies with the type-volume limitation of Fed. R. App. P. 32(a)(7)(B) and 28.1(e)(2)(B)(i). This brief contains 16,396 words, excluding the parts of the brief exempted by Fed. R. App. P. 32(a)(7)(B)(iii) and Federal Circuit Rule 32(b).

2. This brief complies with the typeface requirements of Fed. R. App. P. 32 (a)(5) and the typestyle requirements Fed. R. App. P. 32(a)(6). This brief has been prepared in a proportionally spaced typeface using Word 2010 in 14-point Times New Roman font.

Dated: September 30, 2015

Respectfully submitted,

/s/ *Mark E. Miller*

Mark E. Miller
O'Melveny & Myers LLP
Two Embarcadero Center, 28th Floor
San Francisco, California 94111
Telephone: (415) 984-8700
Facsimile: (415) 984-8701
markmiller@omm.com

Counsel for Plaintiff-Cross-Appellant
Mentor Graphics Corp